Privacy in FL

Chuan Xu

Thread model

Access point	Actor	Thread model
Server	Root access to the server (administrator)	-inspects all messages sent to server -tampers with the training process
Client	Root access to the client device (by design or by compromising the device)	-inspects all messages sent from server -tampers with the training process
Output Models	Engineers & Analysts	may have access to multiple outputs from the system, e.g. sequences of model iterates from multiple training runs with different hyperparameters
Deployed Models	The rest of the world	-black box access -white box access

-Honest but curious
-Malicious

Existing attacks for FL

- Thread model: honest-but-curious server
- Gradient inversion attack
  - Gradient-based: attack directly on the gradients
    [local model - server model]
  - Invert the gradients to recover the whole/partial inputs
  - Image dataset

Figure: 5 most reconginzalbe images when attacking the target client with 100 images from 100 classes in CIFAR100, FedAvg with batch size 100, one local step, ResNet32 [Geiping et al, 2020]

Limitations

Sensitive to the batch size and the local step

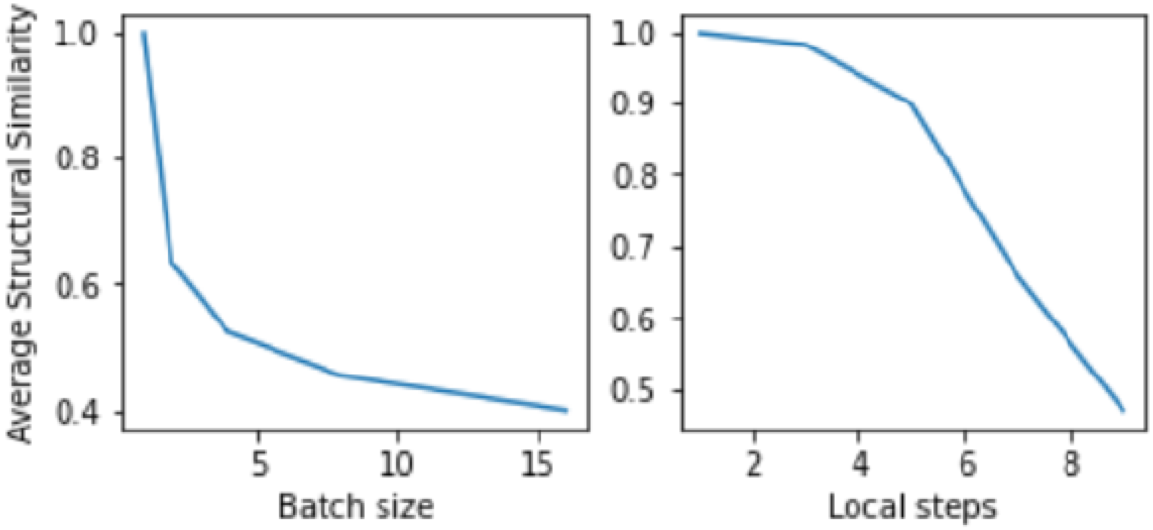


Figure: Average structure similarity between reconstructed and original images in Labeled Faces in the Wild dataset, FedAvg LetNet model. [Wei et al, 2020]

Limitations

- Works well only when the labels of the batch is known to the attacker [Huang et al, 2021]

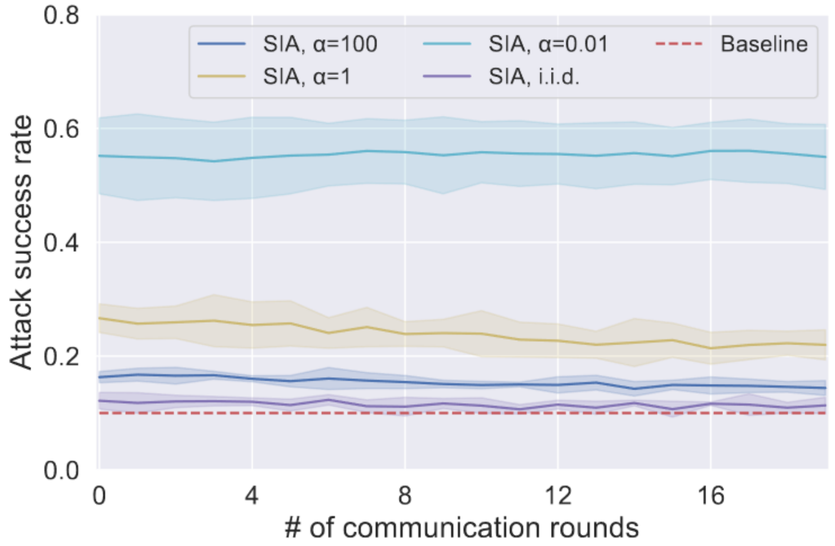
CIFAR10. BATCH SIZE 16

(b) Reconstructions with and without private labels

Existing attacks for FL

- Thread model: honest-but-curious server
- Source inference attack [Hu et al, 2021]
  - ❑Identify a given instance comes from which client
  - ❑Intuition: smaller loss of client k’s local model on a training record z, the higher posterior probability that z belongs to the client k.
  - ❑Works better when local updated models overfits (data is non i.i.d and the number of local steps is large)
  - ❑Ideal case ->

    the local returned model is local optimum



Existing attacks for FL

- Thread model: honest-but-curious server
- Local model reconstruction attack [Xu et al, 2021]

Goal: reconstruct the model θ_c^* a client would have trained using only its own local dataset

$$\boxed{\textbf{local model} \qquad \theta_c^* = \arg\min_{\theta \in \mathbb{R}^d} \mathcal{L}_c(\theta)}$$

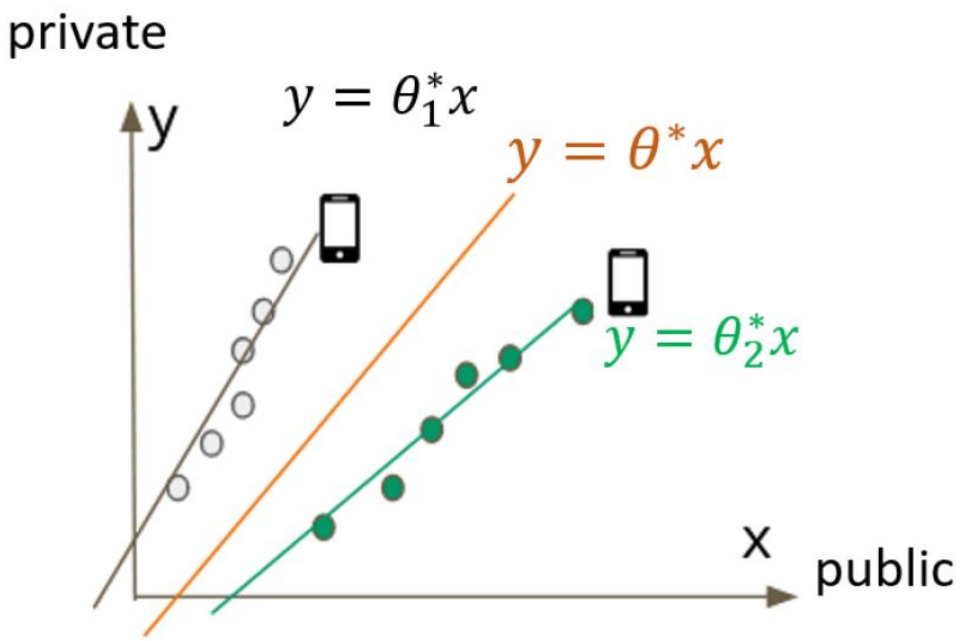


Figure: Infer private information from local model

Adversary collects

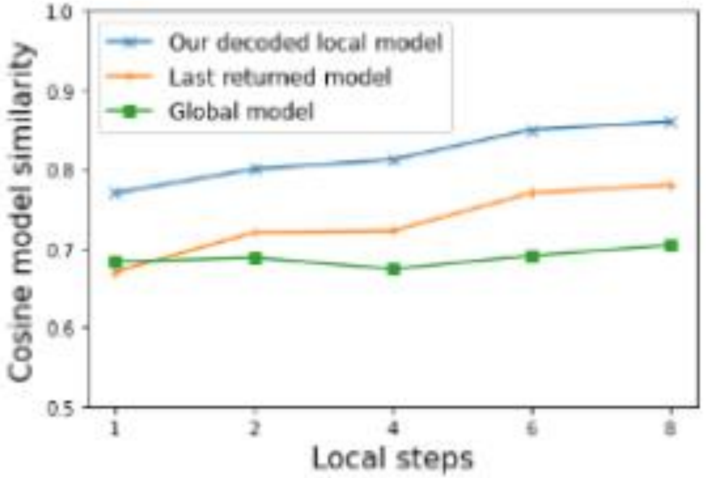
θ_c $\{(\theta, \underbrace{\theta - \theta_c}_{\Delta\theta})\}$

Builds mapping function

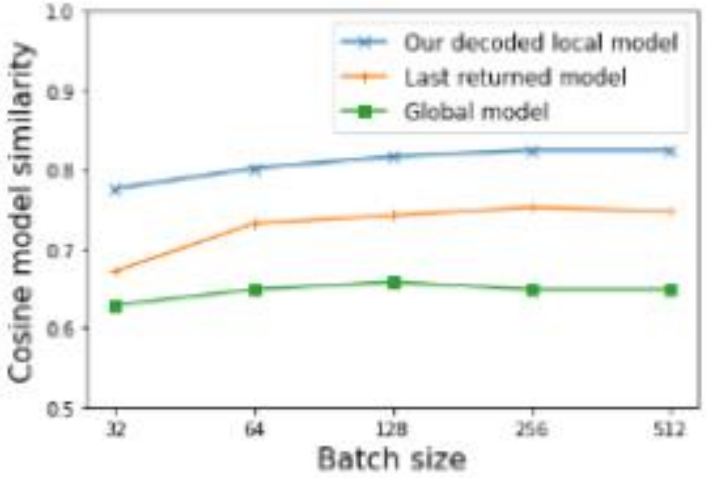
$G^c(\hat{\omega}, \theta) \sim \Delta\theta$

Estimate local model

$\hat{\theta}_c = \min_{\theta} \|G^c(\hat{\omega}, \theta)\|^2$



(c) Adult: Model cosine similarity vs local steps.



(d) Adult: Model cosine similarity vs batch size

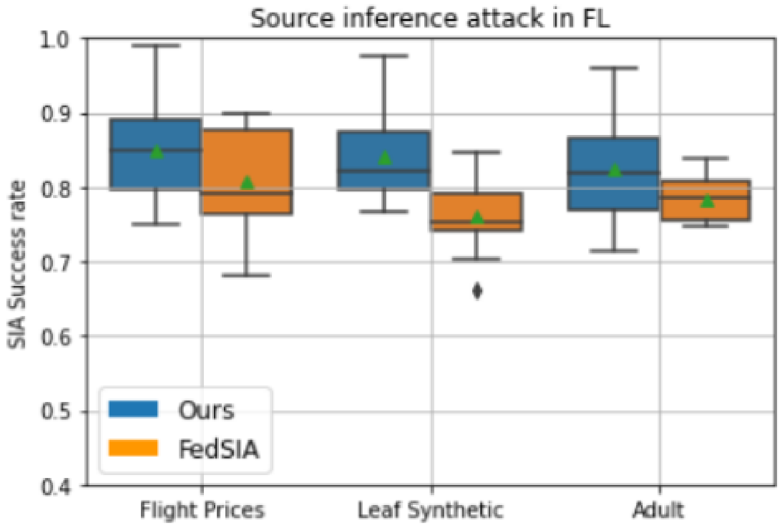
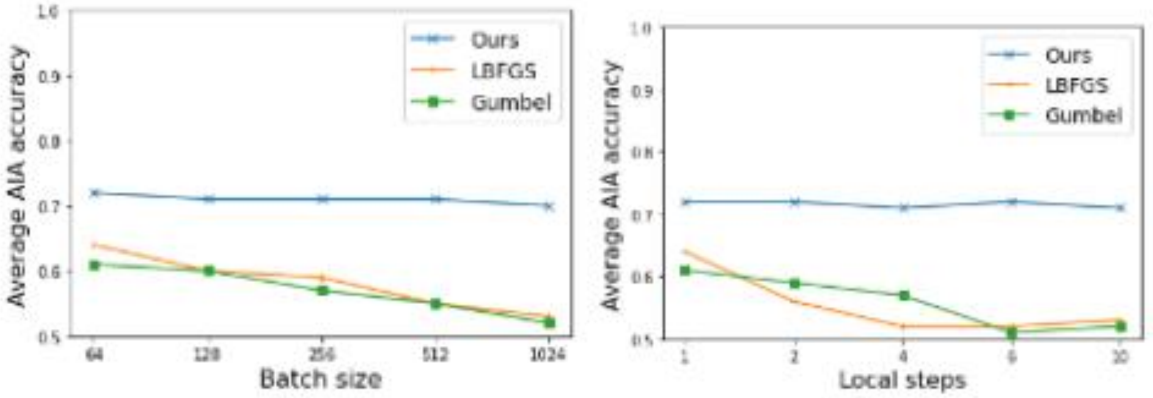


Figure 9: Attack Success rate (ASR) of the source inference attack when training a neural network model (3 hidden layers with 256 neurons per layer) with 10 participating clients, 1 local step and batch size 256.

Existing attacks for FL

- Thread model: honest-but-curious server with additional knowledge
- Attribute inference attack [Lyu et al, 2021], [Driouich et al, 2022]

e.g., health status, political preference, income…

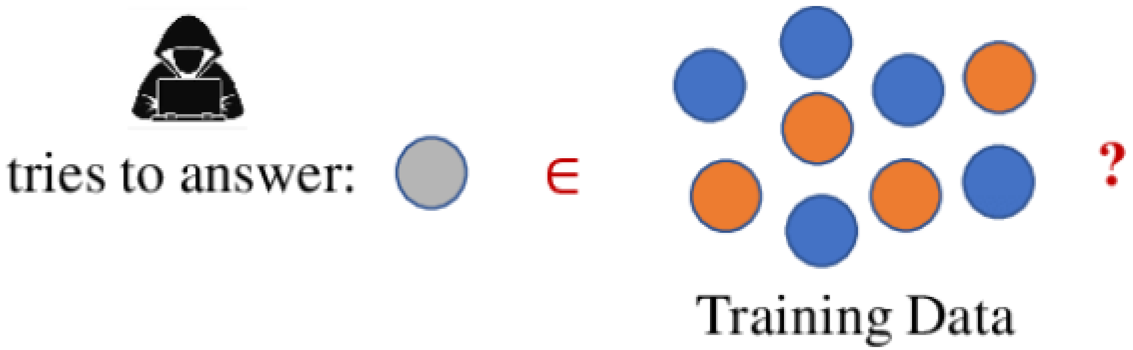


(e) Leaf Synthetic: AIA accuracy vs batch size (1 local step, 10 clients)
(f) Leaf Synthetic: AIA accuracy vs local steps (batch size 256, 10 clients)

Figure 5: Impact of batch size and local steps on AIA performances while training a neural network (3 hidden layers with 256 neurons per layer).

Existing attacks for FL

- Thread model: honest-but-curious server with additional knowledge
- Membership inference attack [Zari et al, 2021]



Thread model: honest-but-curious server

- Gradient inversion attack
- Source inference attack
- Local model reconstruction attack
- Attribute inference attack
- Membership inference attack

Existing attacks for FL

- Thread model: malicious client
- Adversarial attacks: modify the behavior of the model
  - Untargeted attack (reduce the global model accuracy)
  - Targeted attack (e.g., predict a target label τ on any input data that has an attacker-chosen pattern embedded)

- How the adversarial attacks works:
  - Data poisoning
  - Model update poisoning

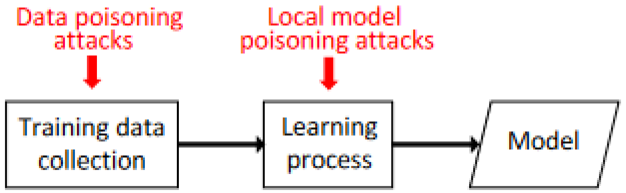
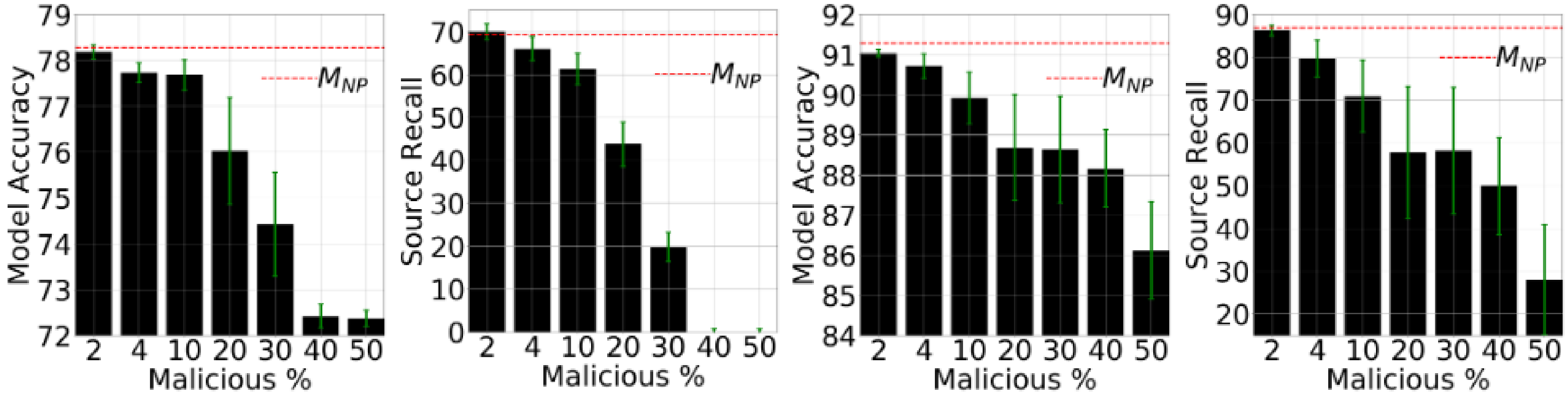


Figure 1: *Data* vs. *local model* poisoning attacks.

Data poisoning: label-filpping [Tolpegin et al, 2020]



(a) CIFAR-10 M^{acc} (b) CIFAR-10 c_{src}^{recall} (c) F-MNIST M^{acc} (d) F-MNIST c_{src}^{recall}

Fig. (1) Evaluation of attack feasibility and impact of malicious participant percentage on attack effectiveness. CIFAR-10 experiments are for the $5 \rightarrow 3$ setting while Fashion-MNIST experiments are for the $4 \rightarrow 6$ setting. Results are averaged from 10 runs for each setting of $m\%$. The black bars are mean over the 10 runs and the green error bars denote standard deviation.

Exisiting attacks for FL

- Thread model: Deployed models
- Model inversion attack [Fredrikson et al, 2015]

Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

Demo: [Colab for centralized case]

Defense

- Thread model: honest-but-curious adversary
- Differential private algorithms
- It provides strong, worst-case protections against a variety of attacks
- Definition of the differential privacy in FL
    - ❑Untrusted server (local differential privacy)
    - ❑Trusted server
        - Sample level
        - Client level

Algorithm 1 Differentially private SGD (Outline)

Input: Examples $\{x_1, \ldots, x_N\}$, loss function $\mathcal{L}(\theta) = \frac{1}{N}\sum_i \mathcal{L}(\theta, x_i)$. Parameters: learning rate η_t, noise scale σ, group size L, gradient norm bound C.

Initialize θ_0 randomly

for $t \in [T]$ **do**

Take a random sample L_t with sampling probability L/N

Compute gradient

For each $i \in L_t$, compute $\mathbf{g}_t(x_i) \leftarrow \nabla_{\theta_t}\mathcal{L}(\theta_t, x_i)$

Clip gradient

$\bar{\mathbf{g}}_t(x_i) \leftarrow \mathbf{g}_t(x_i) / \max\left(1, \frac{\|\mathbf{g}_t(x_i)\|_2}{C}\right)$

Add noise

$\tilde{\mathbf{g}}_t \leftarrow \frac{1}{L}\left(\sum_i \bar{\mathbf{g}}_t(x_i) + \mathcal{N}(0, \sigma^2 C^2 \mathbf{I})\right)$

Descent

$\theta_{t+1} \leftarrow \theta_t - \eta_t \tilde{\mathbf{g}}_t$

Output θ_T and compute the overall privacy cost (ε, δ) using a privacy accounting method.

Trade of between the privacy guarantee and the model utility

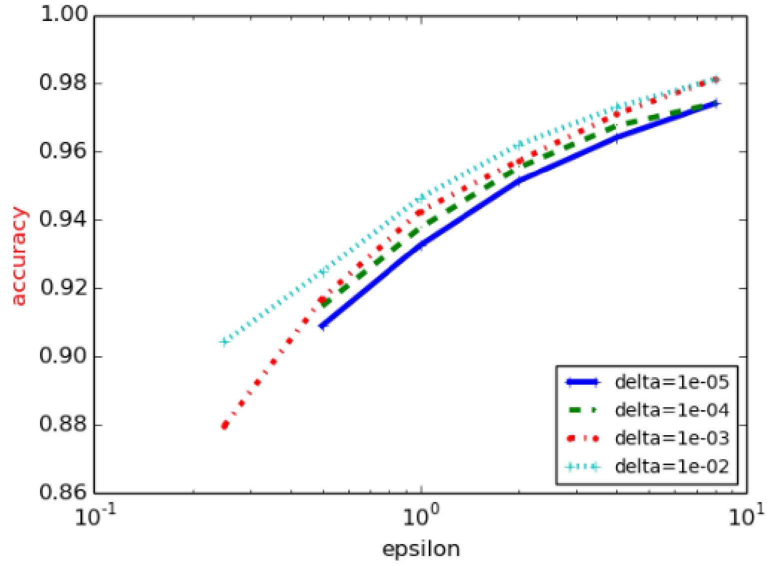
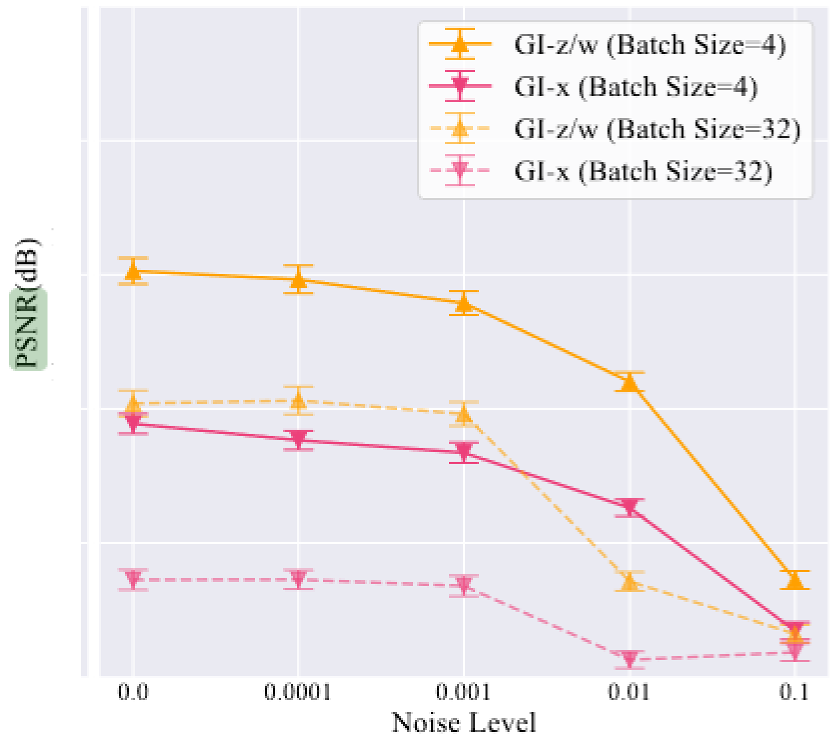


Figure 4: Accuracy of various (ε, δ) privacy values on the MNIST dataset. Each curve corresponds to a different δ value.

[McMahan et al, 2016]

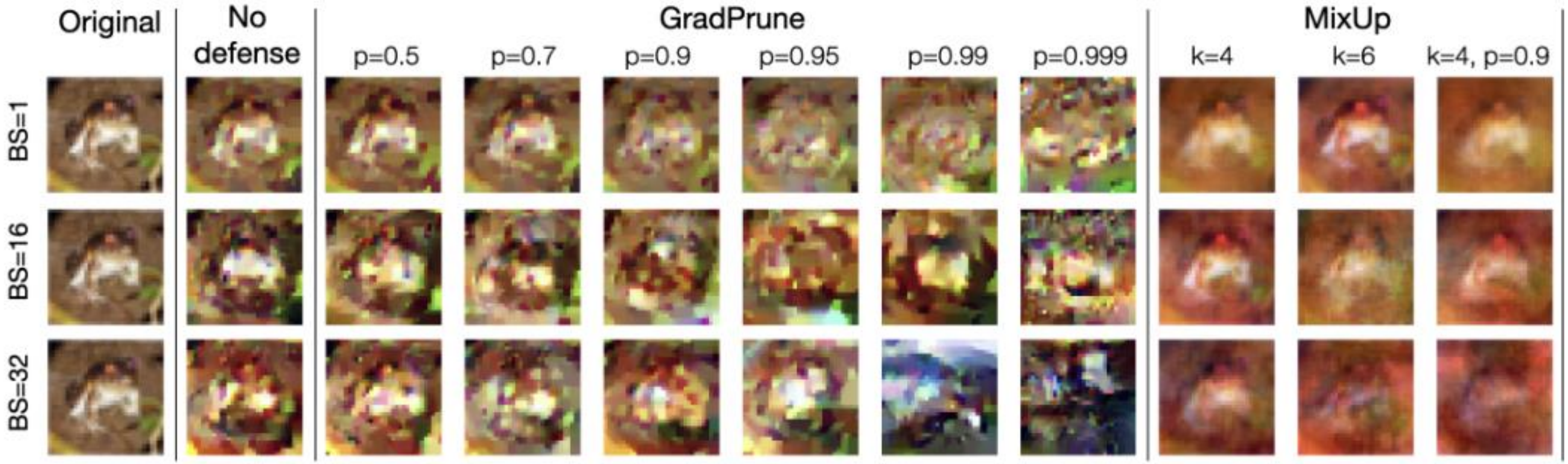
DP defenses against gradient inversion attack [Jeon et al, 2021]



[Peak signal-to-noise ratio (PSNR)]/ Higher indicates the reconstructed image is closer to the original one

Defense

- Thread model: honest-but-curious adversary
- Mixup, gradient/model pruning
- Performance against gradient inversion attack [Huang et al, 2021]



Defense

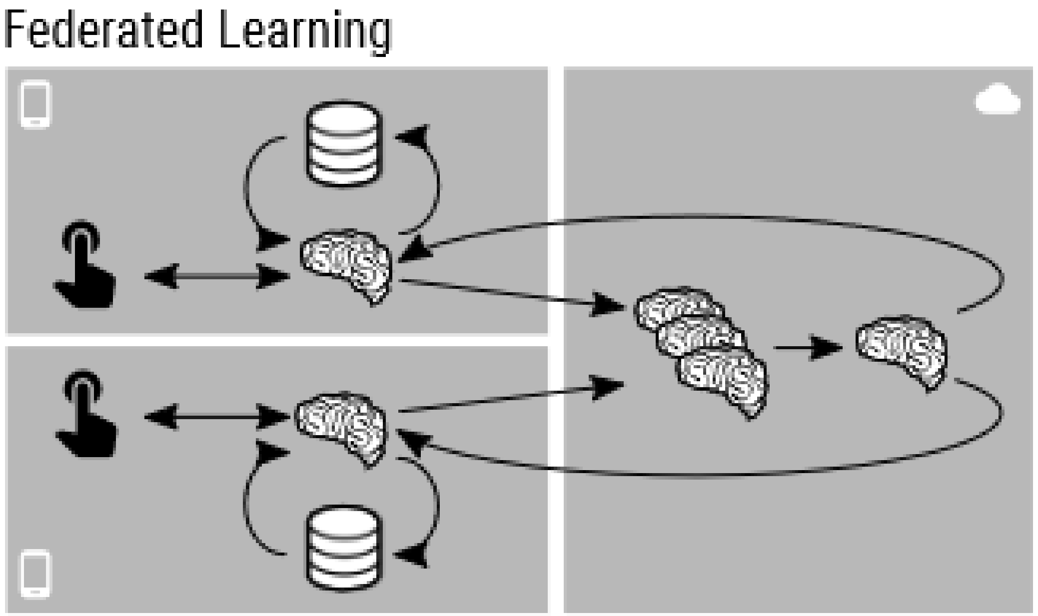
- Thread model: malicious adversary who tampers with the training
- Byzantine resilient algorithm
- Against the worst adversarial attacks where the adversary can cause the process to produce any arbitrary output.
- Basic ideas: replaces the averaging step on the server with a robust estimate of the mean [Blanchard et al, 2017] [Yin et al, 2019]

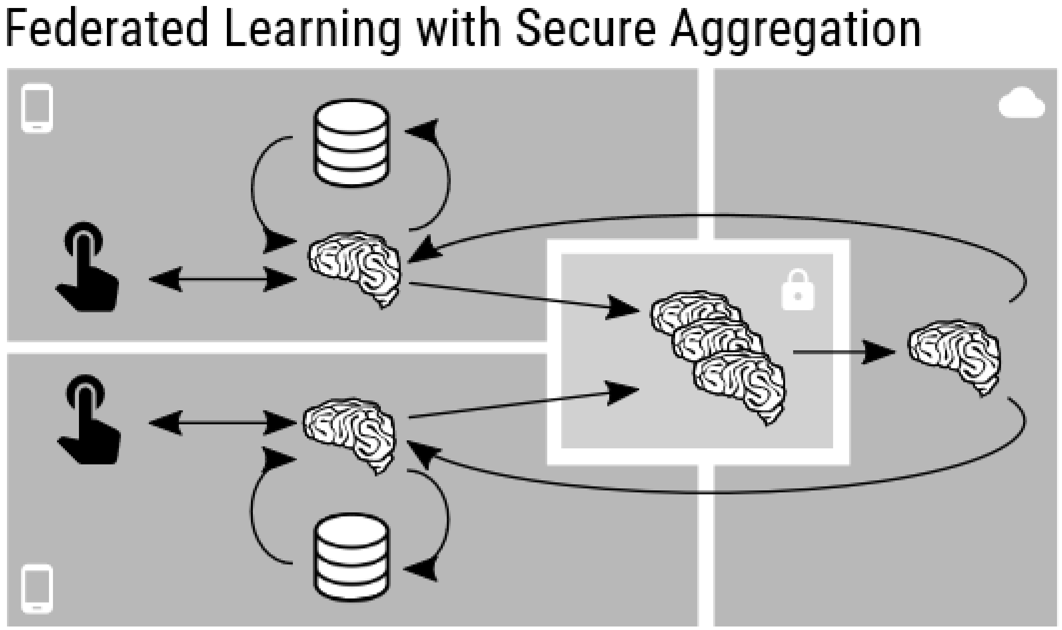
Defense

- Reduce the ability of the adversary :

Secure aggregation [Bonawitz et al, 2017]

  - Thread model: honest-but-curious adversary





Through additive masks, with additional computation load (quadratic for the users)

Defense

- Reduce the ability of the adversary :

Trust Execution Environment (TEE)

  - a secure enclave within a CPU that is protected by embedded encryption keys and authentication mechanisms.
  - (1) Authenticity: the code under execution should not have been changed
  - (2) integrity: runtime states should not have been tampered with
  - (3) confidentiality: code, data and runtime states should not have been observable by unauthorized application

  - An open challenge to implement a reliable TEE platform in FL due to the limited memory and resource infrastructure, and the required processes needed to connect verified codes

References

- [Geiping et al, 2020] Inverting gradients–How easy is it to break privacy in federated learning? NeurIPS 2020
- [Wei et al, 2020] A Framework for Evaluating Gradient Leakage Attacks in Federated Learning, Arxiv
- [Huang et al, 2021] Evaluating Gradient Inversion Attacks and Defenses in Federated Learning, NeurIPS 2021
- [Hu et al, 2021] Source Inference Attacks in Federated Learning, ICDM 2021.
- [Xu et al, 2021] Chuan Xu, Giovanni Neglia. What else is leaked when eavesdropping Federated Learning?. CCS workshop Privacy Preserving Machine Learning (PPML), 2021
- [Lyu et al, 2021] A Novel Attribute Reconstruction Attack in Federated Learning,
- FTL-IJCAI-2021
- [Driouich et al, 2022] A nouvel model-based attribute inference attack in federated learning, FL-NeurIPS22, Dec 2022.
- [Zari et al, 2021] Efficient Passive Membership Inference Attack in Federated Learning, NeurIPS workshop on Privacy in Machine Learning (PriML), 2021
- [Tolpegin et al, 2020] Data Poisoning Attacks Against Federated Learning Systems, ESORICS 2020
- [Fredrikson et al, 2015] Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures
- [McMahan et al, 2016] Deep Learning with Differential Privacy, CCS 2016
- [Jeon et al, 2021] Gradient Inversion with Generative Image Prior, NeurIPS 2021
- [Blanchard et al, 2017], Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent, NIPS 2017
- [Yin et al, 2019]. Byzantine-robust distributed learning:Towards optimal statistical rates. In ICML, 2019.
- [Bonawitz et al, 2017], Practical Secure Aggregation for Privacy-Preserving Machine Learning, CCS 2017