GradSec: a TEE-based Scheme Against Federated Learning Inference Attacks
Aghiles Ait Messaoud, Sonia Ben Mokhtar, Vlad Nitu, Valerio Schiavoni

To cite this version:

HAL Id: hal-03752260
https://hal.archives-ouvertes.fr/hal-03752260
Submitted on 16 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
GradSec: a TEE-based Scheme Against Federated Learning Inference Attacks

Aghiles Ait Messaoud
Ecole Nationale Supérieure d'Informatique, Algeria

Vlad Nitu
University of Lyon, LIRIS, CNRS, France

Sonia Ben Mokhtar
University of Lyon, LIRIS, CNRS, France

Valerio Schiavoni
University of Neuchâtel, Switzerland

1 Introduction

Federated learning (FL) is a distributed machine learning (ML) approach, which attracted attention thanks to its ability of training ML models while keeping raw data under the control of their producer, i.e., end users. However, the long list of privacy attacks (e.g., [11, 13, 18]) prove that the model updates shared in the context of FL training still constitute a threat to users’ privacy. In particular, the model’s gradients [17], generally used to update model parameters, may leak sensitive information enabling for instance the reconstruction of raw data samples or learning hidden properties about the participating users (e.g., their race, gender).

Trusted execution environments (TEEs) [15] are recent turn-key solutions that provide program execution with privacy and confidentiality guarantees (e.g., ARM TrustZone [14], Intel SGX [4], AMD SEV [7]). Typically, TEEs can execute secure enclaves, shielding read and write access to an application’s protected code and data against compromised operating systems, or system libraries. This work provides a secure FL scheme to mitigate inference attacks using TEEs.

Context and threat model. We study three types of inference attacks, all launched by a compromised or malicious FL client: DRIA (Data-Reconstruction Inference Attack) [18], MIA (Membership Inference Attack) [13] and DPIA (Data-Property Inference Attack) [11]. We detail their threat model as defined in the original papers.

(1) DRIA aims at reconstructing original input data based on the emitted model gradients. The attacker is a spyware running in an FL client device, monitoring the FL training process, particularly the gradients produced. It looks for two emitted gradients, with respect to original input and with respect to attacker’s random input, respectively. Then, through an optimisation algorithm similar to Gradient Descent, the attacker minimizes the distance between the two gradients by optimising the random data. At the end of the optimisation process, the attacker manages to get a random data as close as possible to the training data.

(2) MIA’s goal is to learn whether specific data instances are present in the global model training dataset (D). We assume that a malicious FL client has prior knowledge about D, i.e., some of the data part of D (D₁ ⊂ D) and some of the data that aren’t (D₂). The attacker trains a binary classifier (Attack Model) on global model gradients with respect to D₁ and D₂. Further, if the attacker wants to make an inference about membership probability of any data, he feeds it to the global model, gets its gradients, then feed them to his Attack Model. (3) DPIA infers the presence probability of private properties in the input data. As MIA, we assume a malicious FL client trains a binary classifier (Attack Model) on global model gradients with respect to attacker’s auxiliary data, collected along many FL cycles. Then, if the attacker wants to infer the presence probability of a private property among batches of data used to train the global model during an FL cycle, he computes the difference between two consecutive snapshots of the global model to get the aggregated gradients, and fed them to the Attack Model.

In all the previous threat models, the FL Server is assumed to be a honest entity that uses Secure Aggregation [2] to avoid spying on individual FL client gradients, preventing privacy attacks from him self. All threat models assume an honest-but-curious attacker not interfering with the normal FL process and message exchanges. Finally, we assume the used ML models are exclusively feed-forward neural networks [5] (e.g., fully-connected and convolutional ones [1]) trained by stochastic gradient descent [3], a popular optimization for feed-forward networks.

Goals. We propose GradSec, a TEE-based gradient protection mechanism for FL architectures. Intuitively, reducing the amount of gradients accessible from the model will reduce the accuracy of inference attacks. However, storing the optimization process for the entire FL model into an enclave will introduce large overheads and increase the attack surface. Unlike previous approaches (i.e., DarkneTZ [12]), GradSec can protect non-successive layers of the FL model, a strategy which can substantially reduce the overheads while providing similar levels of protection against attacks.

2 GradSec: architecture and workflow

GradSec is a TEE-based scheme that aims at securing model gradients during the FL training. Our design is driven by two main observations on existing neural network systems: (i) An attacker can compute the difference between two consecutive snapshots of a model to deduce the gradients. (ii) An attacker can follow the back-propagation computation flow when the gradients are naturally emitted during training.
GRADSec protects the model parameters and the operations required for the gradient computations of a layer. It supports two execution modes. In static mode, we fix in advance a subset of layers to be protected in the TEE enclave during all the FL cycles. This approach is similar to DarkneTZ [12]. We overcome DarkneTZ’s limitation by implementing the ability to protect non-successive layers inside the TEE enclave. In the dynamic mode the protected layers can change from one FL cycle to another by leveraging a moving window \( MW \), defined by two parameters: its size \( size_{MW} \) (the number of successive layers protected in the TEE at a time) and its probability distribution \( V_{MW} \) (the probability that the window protects a specific set of layers).

DRIA and MIA are single-shot attacks. Hence, an attacker only needs one iteration of model training to get the gradients needed for the attack model. Therefore, for such attacks only GRADSec static mode can be effective. Instead, the DRIA attack is carried out over multiple FL cycles, giving dynamic GRADSec sufficient time and opportunities to vary the protected layers.

As shown in our preliminary evaluation results, we evaluated the efficiency of the static mode against all the considered attacks (DRIA, MIA and DPIA), limiting instead the dynamic mode to the DPIA.

3 Preliminary Evaluation

We consider two distinct training models and real-world datasets. We launched DRIA and MIA against the model LeNet in [9] (4 convolutional layers and 1 fully connected layer) and CIFAR-100 [8]. We rely on the DPIA official implementation [16] (3 convolutional layers and 2 fully connected layers) using the LFW dataset [6].

We measure the performance of DRIA via the ImageLoss metric, i.e., the euclidean distance between the attacker’s inferred image and the original FL client image fed to the model. We measure the performance of MIA and DPIA using AUC, i.e., an aggregated measure of the attack model performance considering all the possible classification thresholds. It is statistically consistent and more discriminating measure than accuracy [10]. An attack model with an AUC of 0.5 is considered as inefficient and similar to a random guess.

**DRIA.** Securing early layers (especially the 2nd layer) with static GRADSec is sufficient to make the attacker get a completely blurry reconstructed image, thus a big ImageLoss like shown in Figure 1. It is statistically consistent and more discriminating measure than accuracy [10]. An attack model with an AUC of 0.5 is considered as inefficient and similar to a random guess.

**MIA.** Securing tail layers (i.e., the 5th layer) with static GRADSec significantly lowers the attack AUC from 0.96 to 0.85. Protecting more layers show little benefits, as the AUC attack only reaches 0.80 with last 4 layers protected as shown in Table 1.

**DPIA.** Protecting individual layers using static GRADSec proves barely effective against this attacks. We just manage to hit an AUC rate of 0.88 by protecting the 4th layer. While it was possible to lower the AUC down to 0.70 with 4 protected layers inside the enclave, protecting 4 layers uses a lot of secure memory, a scarce resource shared with other Secure Applications. However, GRADSec in dynamic mode is able to achieve the same AUC rate (0.70) with only two simultaneous layers inside the enclave \( size_{MW} = 2 \). Finally, we manage to lower the AUC further to 0.64 and 0.62 with \( size_{MW} = 3 \) and \( size_{MW} = 4 \) respectively. These results are resumed in Tables 2 and 3.

**Grouped protection.** Given its ability to protect non-successive layers, GRADSec is able to simultaneously minimize the impact of both DRIA and MIA, without the need to protect all intermediate layers as required in DarkneTZ. By only protecting the 2\(^{nd} \) and 5\(^{th} \) layer, GRADSec can reduce the memory footprint of DarkneTZ by 10% while providing similar levels of protection.

4 Conclusion and Future Work

We presented GRADSec, a TEE-based protection mechanism that improves the FL privacy guarantees. GRADSec can operate in two modes: static and dynamic. Static GRADSec can simultaneously protect against DRIA and MIA attacks while dynamic GRADSec is able to increase the protection against DPIA.

References


