GradSec: a TEE-based Scheme Against Federated Learning Inference Attacks

Aghiles Ait Messaoud, Sonia Ben Mokhtar, Vlad Nitu, Valerio Schiavoni

GradSec: a TEE-based Scheme Against Federated Learning Inference Attacks

Aghiles Ait Messaoud
Ecole Nationale Supérieure d'Informatique, Algeria

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

Vlad Nitu
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 ($D_1 \subset D$) and some of the data that aren't (D_2). The attacker trains a binary classifier (Attack Model) on global model gradients with respect to D_1 and D_2. 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 DPIA 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) using 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 infered 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 2*nd* 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.

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.

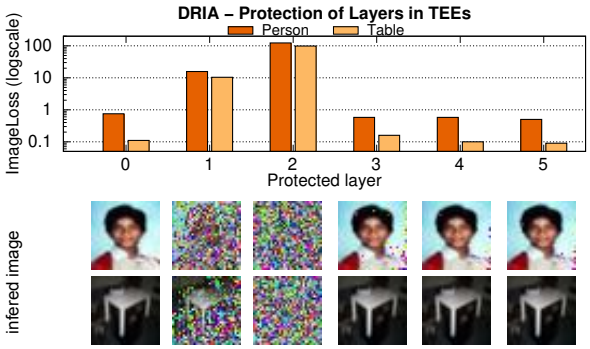

Figure 1. ImageLoss of two infered images.

	None	L5	L5+L4	L5+L4+L3	L5+L4+L3+L2
AUC	0.95	0.85	0.84	0.82	0.8

Table 1. AUC of MIA with various protected layers.

	None	L4	L3+L4	L2+L3+L4	L1+L2+L3+L4
AUC	0.94	0.88	0.83	0.79	0.70

Table 2. AUC of DPIA using Static GradSec

	None	MW=2	MW=3	MW=4
AUC	0.94	0.71	0.64	0.62

Table 3. AUC of DPIA using Dynamic GradSec

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

[1] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee, 2017.

[2] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017.

[3] Léon Bottou. Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer, 2012.

[4] Victor Costan and Srinivas Devadas. Intel sgx explained. *IACR Cryptol. ePrint Arch.*, 2016(86):1–118, 2016.

[5] Terrence L Fine. *Feedforward neural network methodology*. Springer Science & Business Media, 2006.

[6] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[7] David Kaplan, Jeremy Powell, and Tom Woller. Amd memory encryption. *White paper*, 2016.

[8] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[9] Song Han Ligeng Zhu, Zhijian Liu. https://colab.research.google.com/gist/Lyken17/91b81526a8245a028d4f85ccc9191884/deep-leakage-from-gradients.ipynb#scrollTo=AorI020iVjjS, 2019.

[10] Charles X Ling, Jin Huang, Harry Zhang, et al. Auc: a statistically consistent and more discriminating measure than accuracy. In *Ijcai*, volume 3, pages 519–524, 2003.

[11] Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE, 2019.

[12] Fan Mo, Ali Shahin Shamsabadi, Kleomenis Katevas, Soteris Demetriou, Ilias Leontiadis, Andrea Cavallaro, and Hamed Haddadi. Darknetz: towards model privacy at the edge using trusted execution environments. In *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, pages 161–174, 2020.

[13] Milad Nasr, Reza Shokri, and Amir Houmansadr. 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)*, pages 739–753. IEEE, 2019.

[14] Sandro Pinto and Nuno Santos. Demystifying arm trustzone: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 51(6):1–36, 2019.

[15] Mohamed Sabt, Mohammed Achemlal, and Abdelmadjid Bouabdallah. Trusted execution environment: What it is, and what it is not. In *2015 IEEE Trustcom/BigDataSE/ISPA*, volume 1, pages 57–64, 2015.

[16] Congzheng Song. https://github.com/csong27/property-inference-collaborative-ml, 2018.

[17] Hongxu Yin, Arun Mallya, Arash Vahdat, Jose M Alvarez, Jan Kautz, and Pavlo Molchanov. See through gradients: Image batch recovery via gradinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16337–16346, 2021.

[18] Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients, 2019.