2023 - ESS - **61A-BC8** - Securing Data Privacy of Machine-learning Models on Edge Devices using Trusted Execution Environment - gramshan@purdue.edu - Gowri Ramshankar

CERIAS

The Center for Education and Research in Information Assurance and Security

Securing Data Privacy of Machine-learning Models on Edge Devices using Trusted Execution Environment

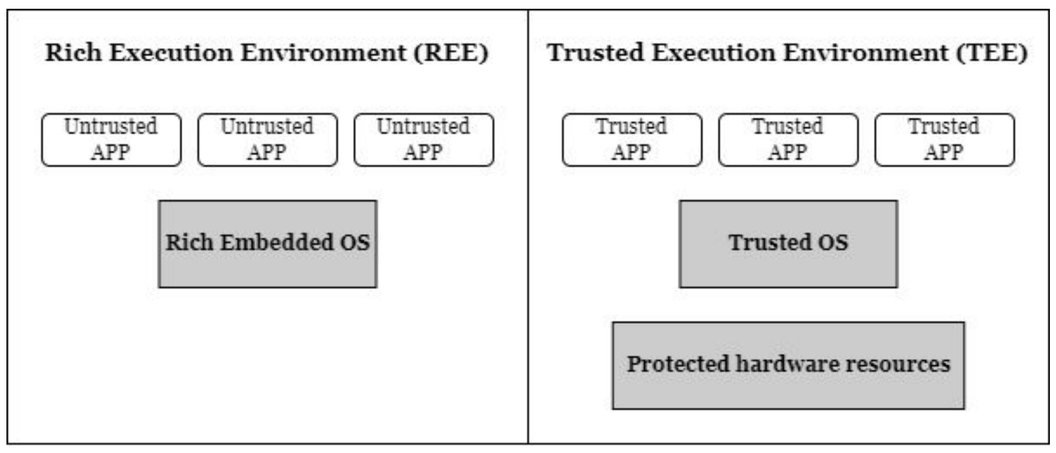
Gowri Ramshankar, Cheng-Yun Yang, Yung-Hsiang Lu

Background

- Machine learning models are under high privacy risk when a large amount of sensitive data is used for training. For example, some business organizations apply machine learning models to analyze the preference of customers based on their private information or past purchase records.
- **Membership inference attacks (MIAs)** are designed to attack such machine learning models. They take the predictions or gradients as input to determine if a specific data is part of the model's training set. If a machine learning model is not well protected during inference, it will result in a private data leakage under MIAs.
- Differential privacy and encryption are two common ways to protect model from MIAs. However, **differential privacy comes with accuracy drop** and **encryption significantly increases the computational overhead**. For edge devices that only have constrained resources (power, memory, computing), we consider Trusted Execution Environment (TEE) as a better choice to secure data privacy.

Trusted Execution Environment (TEE)

- TEE is an area on the main processor of a device that is separated from the system's main operating system. Only trusted applications running on TEE have complete access to the main processor and the **protected hardware resources**. We apply **OP-TEE** on Arm chip with support of **TrustZone** technology as our implementation of TEE.



Problems of using TEE to protect data privacy

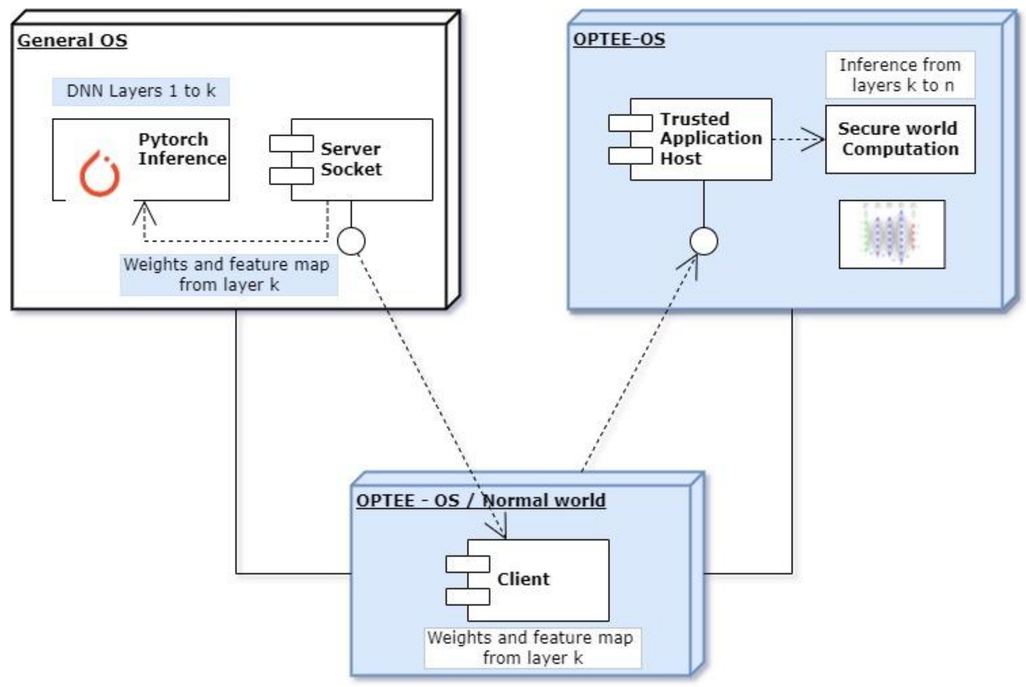
- The protected hardware resources in a TEE are constrained. We are not allowed to put a machine learning model with a large amount of parameters into TEE. Our experiment shows that the largest number of computations we can put in trusted app using OP-TEE is **0.025 BLOPs**.
- The computational cost in TEE is relatively high compared with that in REE because OP-TEE OS only enables CPU. For example, a single image inference on Darknet19 image classification model takes **23.57 seconds** to run using OP-TEE, which is ten times more than that of running on general OS of the same embedding board that has GPU enabled.

Approach

- Our approach is to split the inference process of a model between an inference engine running on a General Purpose Operating System (like Windows) and the Trusted Environment. It is advantageous when compared to existing methods in the following ways:

Existing Methods	Advantages	Disadvantages
DarkNetZ	• Low overhead • Low drop in accuracy • Can be used for Edge Computing	• Does not support Python executables • Can only employ small models
Gramine	• Low overhead • Easy to use	• Not suitable for embedded devices
Our Method	• Suitable for embedded devices • Low – zero drop in accuracy	• Overhead exists because of communication

- Suppose we have am inference model with **n layers** and the weights and feature maps in the last few layers starting from ***k (k<n)*** is deemed to contain the most important information about the inference input. Then, we split the inference of the model between the Normal OS and the Trusted OS as shown below:



References

- OP-TEE: https://optee.readthedocs.io/en/latest/index.html
- Mo, Fan, et al. "Darknetz: towards model privacy at the edge using trusted execution environments." Proceedings of the 18th International Conference on MobiSys, 2020.
- Shokri, Reza, et al. "Membership inference attacks against machine learning models." IEEE symposium on security and privacy, 2017.

PURDUE UNIVERSITY

CERIAS