GuaranTEE: Towards private and attestable ML with CCA

Sandra Siby, Sina Abdollahi, Mohammad Maheri, Marios Kogias, Hamed Haddadi

EuroMLSys, 22 April 2024
On-device Machine Learning

Benefits

- No user data sent to cloud
- Private personalisation
- Latency improvements

- Vision
  - Build features that can process and analyze images and video using computer vision.
- Natural Language
  - Process and make sense of text in different ways, like embedding or classifying words.
- Speech
  - Take advantage of speech recognition and saliency features for a variety of languages.
- Sound
  - Analyze audio and recognize it as a particular type, such as laughter or applause.
On-device Machine Learning

Benefits

No user data sent to cloud

Private personalisation

Latency improvements

Model providers want:

- Model privacy
- Model verifiability and attestability
Protecting ML models

Existing solutions

- Watermarking
  - Detection rather than prevention
  - Evasion attacks

- Cryptography-based
  - Computational and communication overheads

- Hardware-assisted
Protecting ML models

Existing solutions

- Watermarking
  - Detection rather than prevention
  - Evasion attacks

- Cryptography-based
  - Computational and communication overheads

Hardware-assisted

Applications run within Trusted Execution Environments (TEE)
Protecting ML models

Existing solutions

Watermarking
- Detection rather than prevention
- Evasion attacks

Cryptography-based
- Computational and communication overheads

Hardware-assisted
- Mainly tailored to the cloud
- Memory limitations on edge
Protecting ML models

Existing solutions

Hardware-assisted
- Mainly tailored to the cloud
- Memory limitations on edge

Arm’s TEE solutions

Arm’s TrustZone is widely deployed on edge devices.

We consider Arm’s next generation of TEE solutions (deployment expected in 2028):

Confidential Computing Architecture (CCA)
Arm TrustZone

Normal World

EL0
NW VM
App
App

EL1
NW OS

EL2
Hypervisor

EL3
Secure Monitor

Secure World

T App
T App

TOS

SPM
Arm CCA

Realm world

EL0
Realm VM
Realm VM

EL1
Realm VM
NW VM
App
App
T App
T App

EL2
RMM
NW OS
TOS
SPM

EL3
Secure Monitor

Normal World

Root world

Secure World

Hypervisor

Normal World

Secure World
CCA and ML deployment

Why is CCA a promising choice for ML deployment?

- Flexible memory allocation
- Protection against compromised hypervisor
- General-purpose development
CCA and ML deployment

Why is CCA a promising choice for ML deployment?

- Flexible memory allocation
- Protection against compromised hypervisor
- General-purpose development

GuaranTEE
Framework for ML models to be run on end devices in a private and verifiable manner
System overview

Model provider

Realm world

Normal world

Client (Device)

Trusted verifier
System overview

Model provider

Client (Device)

Trusted verifier

Realm world
Normal world
System overview

Model provider

Client (Device)

Trusted verifier

Shared folder

Realm VM

Realm world

Normal world

1

2
System overview

Model provider

Client (Device)

Trusted verifier

Realm VM

Realm world

Normal world

Shared folder

1

2

3

4
System overview

Model provider

Realm provider

Shared folder

Realm VM

Realm world

Normal world

Client (Device)

Trusted verifier
System overview

Model provider

Client (Device)

Trusted verifier

Realm VM

Realm world

Normal world

Shared folder

1

2

3

4

5

6
System overview

1. Trusted verifier
2. Realm world
3. Realm VM
4. Normal world
5. Shared folder
6. Model provider
7. Client (Device)

Model provider

Client (Device)

Trusted verifier
Implementation

Fixed Virtual Platform (FVP)

TensorFlow Lite image recognition model (16 MB)

CCA integration with Secure monitor, RMM, and hypervisor

Shared folder for model inputs and outputs

Applications in normal world and realm
Preliminary evaluation

What we measure: Overhead of inference and realm VM creation over a normal world VM.
Preliminary evaluation

What we measure: Overhead of inference and realm VM creation over a normal world VM.

How we measure: Number of instructions as FVP is not cycle-accurate

• Approximate counting of instructions.
• In progress: implementing Module Trace Interface for exact instructions.
Preliminary evaluation

What we measure: Overhead of inference and realm VM creation over a normal world VM.

How we measure: Number of instructions as FVP is not cycle-accurate

- Approximate counting of instructions.
- In progress: implementing Module Trace Interface for exact instructions.

Main findings

- On average, realm inference takes 1.6x the instructions normal world.
  - Larger number of context switches
  - Realm creation depends on the size of the image.
Preliminary evaluation

**What we measure:** Overhead of inference and realm VM creation over a normal world VM.

**How we measure:** Number of instructions as FVP is not cycle-accurate

- Approximate counting of instructions.
- In progress: implementing Module Trace Interface for exact instructions.

**Main findings**

- On average, realm inference takes 1.6x the instructions normal world.
  - Larger number of context switches
  - Realm creation depends on the size of the image.

**Note:** Full attestation report could not be implemented due to FVP limitations
Considerations for ML deployment with CCA

- Attacks to data pipeline
- Multiple providers on the same device
- Policy enforcement
- Availability guarantees
Summary

• We propose GuaranTEE — a framework using CCA to deploy ML models on end devices in a private and trusted manner.

• We implement GuaranTEE using FVP, and perform a preliminary evaluation.

• We provide future directions and recommendations on ML deployment with CCA.

Code (with a setup guide): https://github.com/comet-cc/GuaranTEE

Get in touch: s.siby@imperial.ac.uk