# IIJ Lab Internship 2025 参加報告

M2 川﨑秀昌

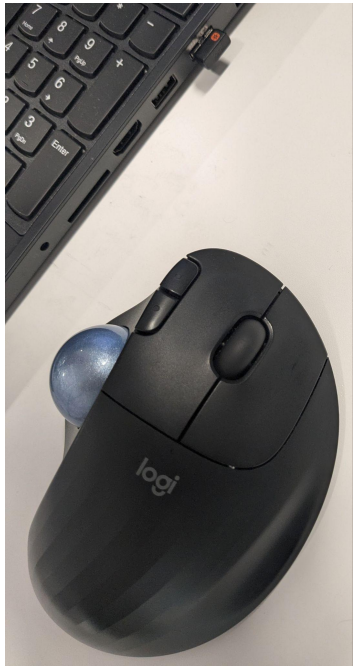
# Securing Human Interface Devices

Kawasaki Hidemasa and Pierre-Louis Aublin

IIJLab summer camp, September 2025

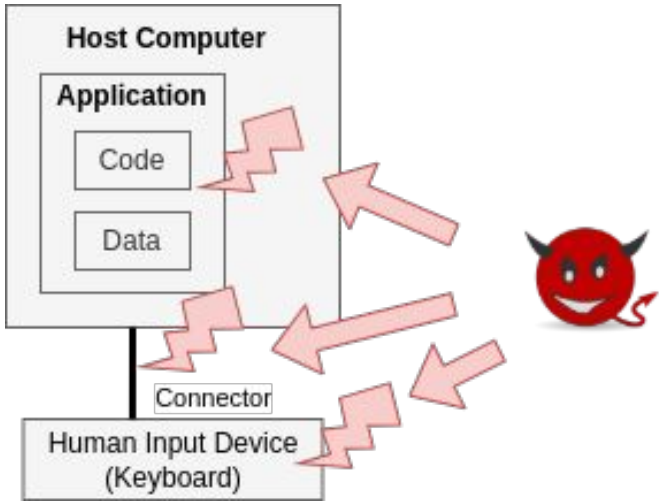
# Human Input Device (HID)

- Keyboard and Mouse etc… send data to host computer



# Security Problem for HID

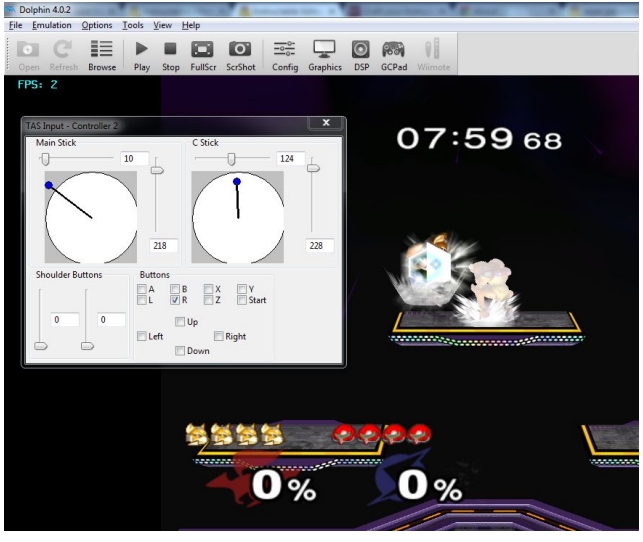
- Attacker has full access to the hardware and software
  - modify application code and data
  - intercept data between HID and computer
  - upload a new firmware to HID
- Attacker can not:
  - observe the key press directly



# Why is unsecure HID device a problem?

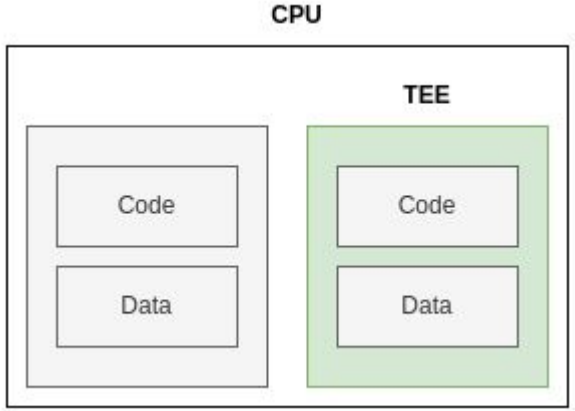
1. Enterprise setting
    a. Keylogger, bad USB cable to leak secrets

2. Video game setting
    a. Aimbot, program to cheats that send fake inputs



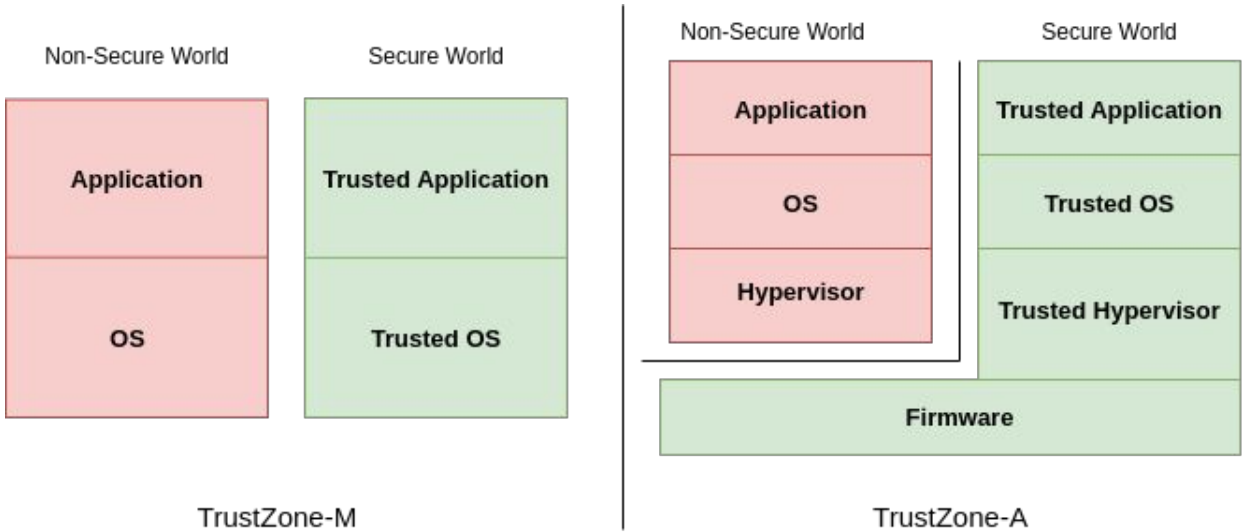
# Trusted Execution Environment (TEE)

- trusted, tamper-proof component in the CPU.
- provides security guarantees of code and data inside
  - confidentiality
  - integrity
- secure against attacker with access to both SW and HW
- implementations:
  - Intel SGX, ARM TrustZone, RISC-V Keystone



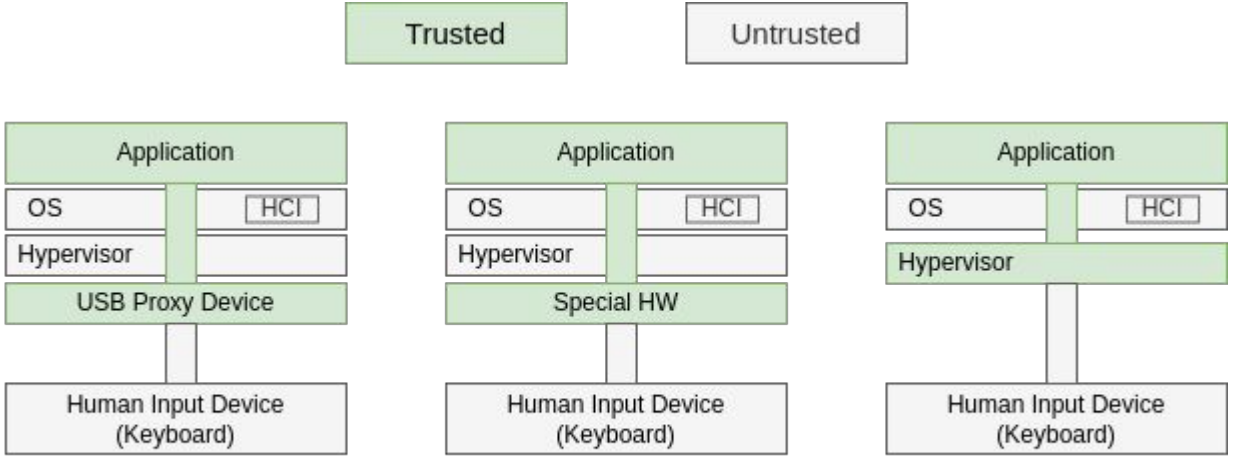
# ARM TrustZone

- Two type of TrustZone:
  - TrustZone-M: for embedded devices
  - TrustZone-A: for rich computing resource devices



# Related Works

- TIO / SGX-USB: Need special trusted USB proxy device
- Aurora: Need special trusted hardware on host motherboard
- SGXIO: Need trusted hypervisor



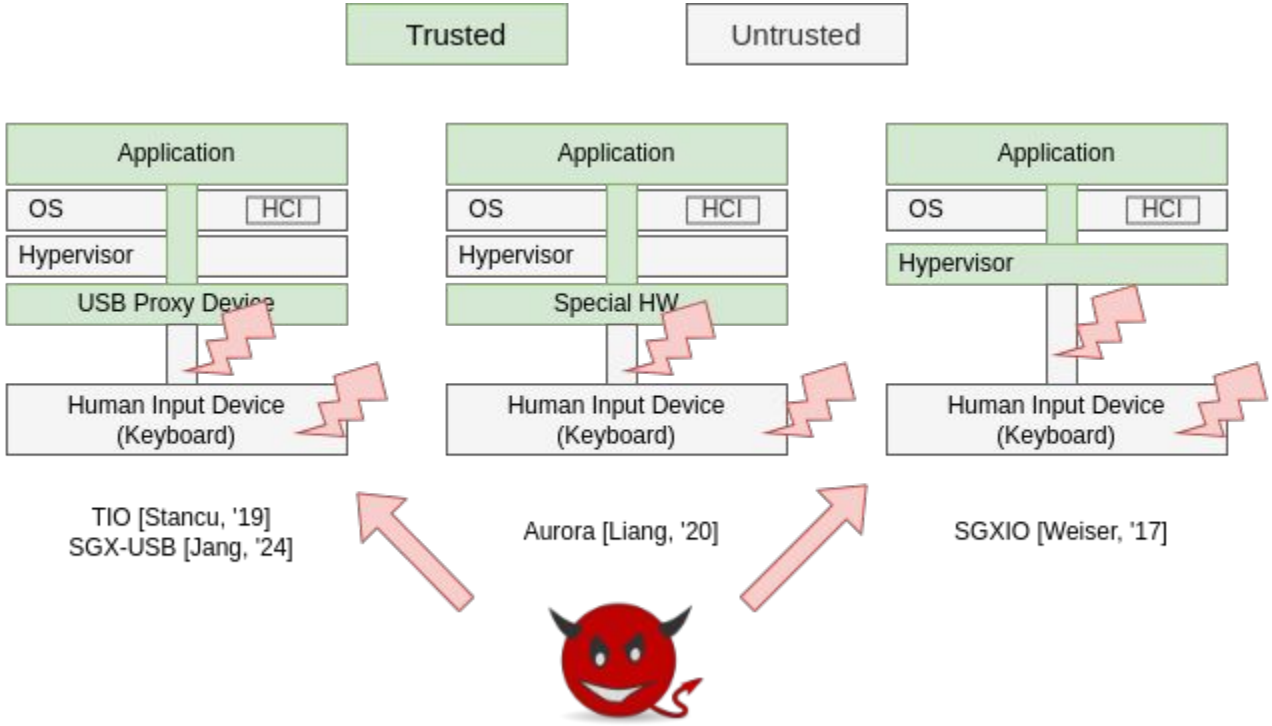

TIO [Stancu, '19]
SGX-USB [Jang, '24]

Aurora [Liang, '20]

SGXIO [Weiser, '17]

# Problem of Existing Related Works

- Attacker is able to tamper with the connector and HID



# Threat Model

- TEE is correctly implemented
  - We trust the TEE
- HID is tampered against intrusions
  - Attacker can’t observe the key press by directly looking into or modifying keyboard
- Attacker has full access to the hardware and software
  - intercept data between HID and computer
  - upload malicious firmware to HID

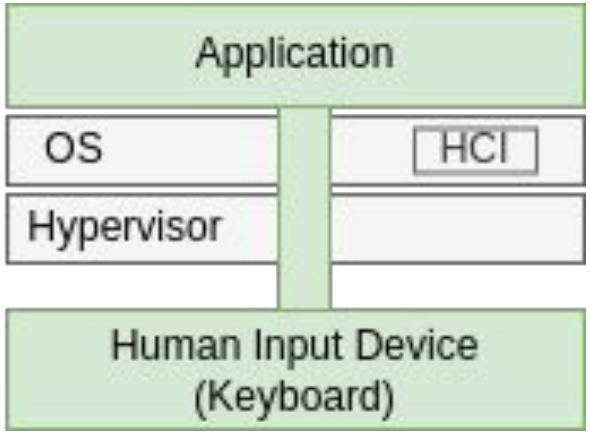
# Challenges

- Ensure confidentiality and integrity at connector and device.
- Attest the device’s firmware and device itself.
- Minimum hardware and software modifications.

# Overview

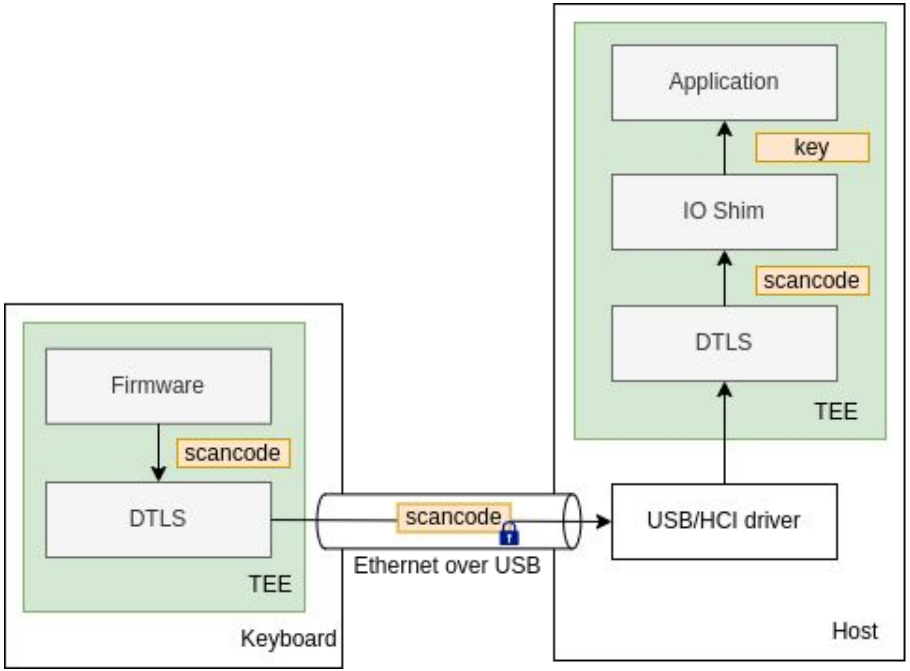
**Key Idea**: End-to-End trusted connection between Application and HID

- Only need to trust application and HID



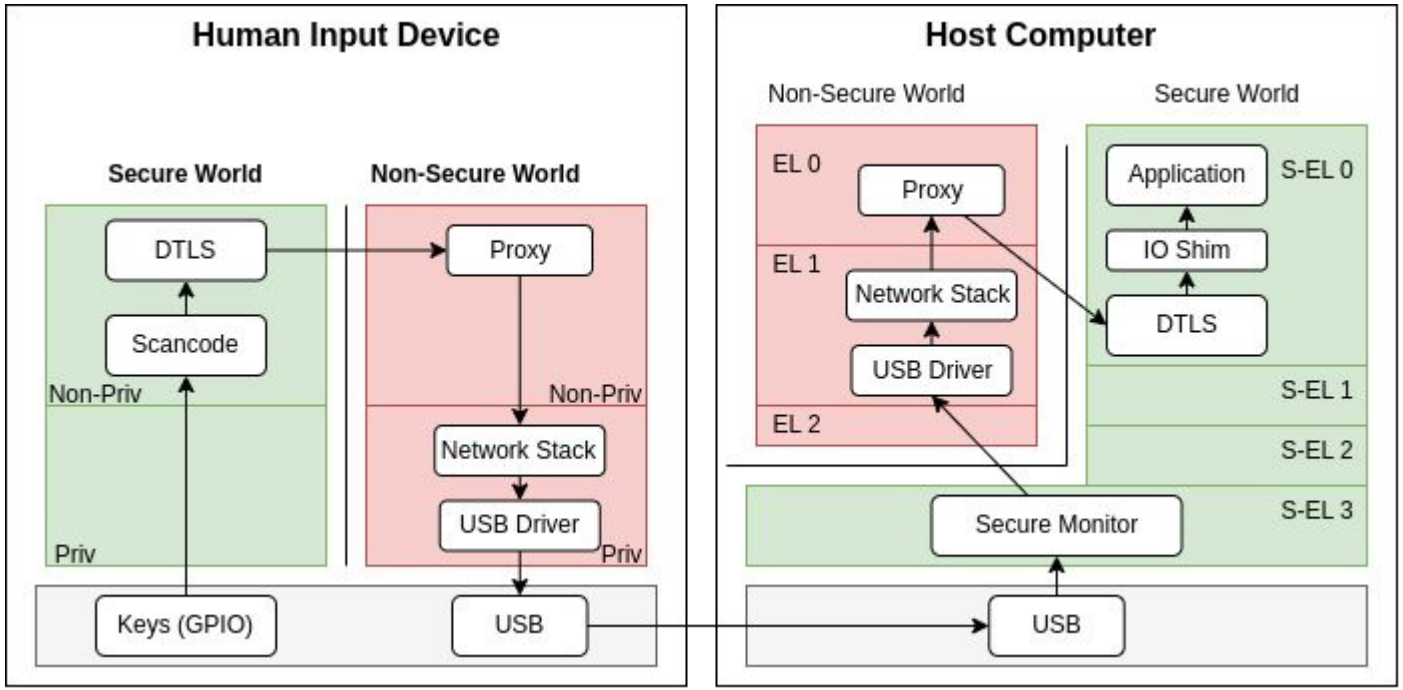
# Architecture

- **Host:**
  - Application, IO Shim and DTLS in TEE
- **HID (Keyboard):**
  - Firmware and DTLS mechanism in TEE
- **Connection:**
  - DTLS ensures confidentiality and integrity of data



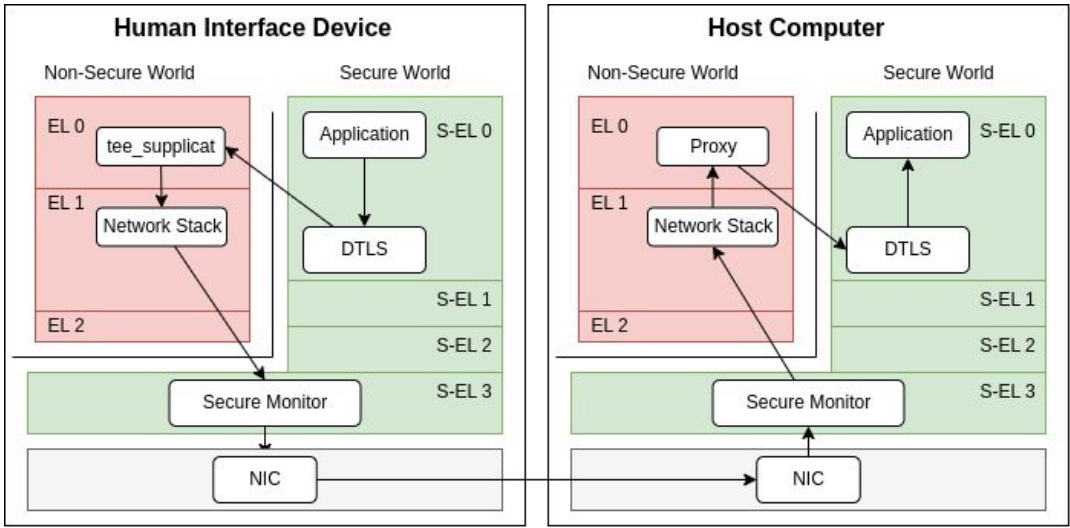
# Ideal Implementation

- TrustZone-M for HID, TrustZone-A for Host



# Our Implementation

- OP-TEE
  - I did: port existing code to OP-TEE + DTLS library (MbedTLS)
  - Challenges in doing so: **Build Server in TEE**
- Raspberry Pi 3B V1.2
- TrustZone-A on both sides
- Use ethernet

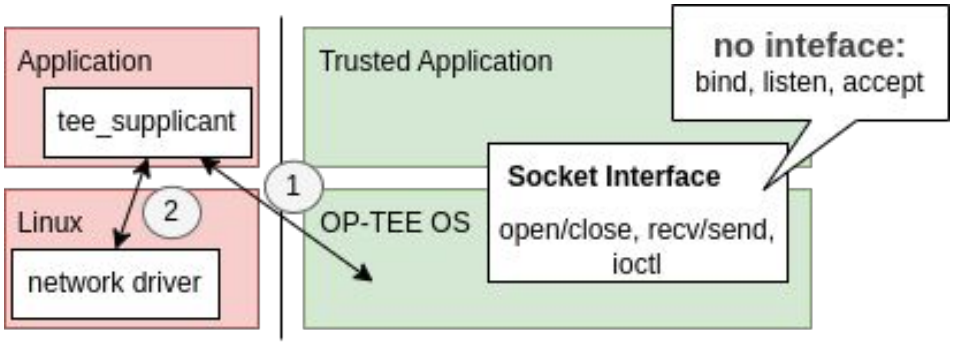


# Implementation Challenges (1/2)

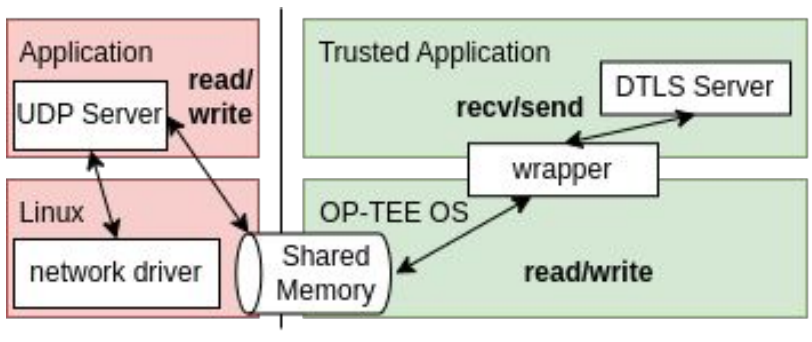
Trusted Application in OP-TEE could not be a UDP/DTLS server.

- No network access from Secure World
- No interface to make socket for server

To enable to make the UDP/DTLS server,  I designed a REE-side Proxy Model.




Implementation Problems



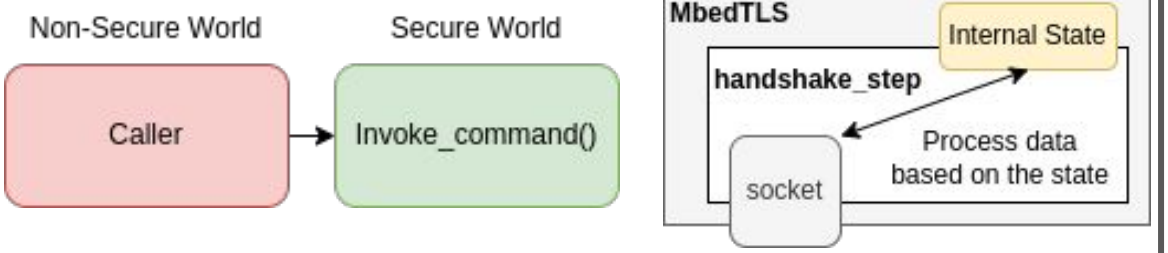

Implementation Details

# Implementation Challenges (2/2)

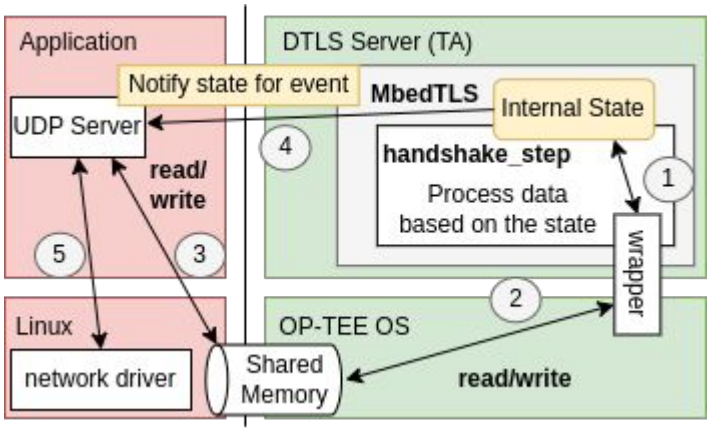
DTLS handshake of MbedTLS library use own internal state to manage the step execution.

- Trusted Application has only one entrypoint from Non-Secure world.
- MbedTLS has internal state for DTLS handshake.

To enable to connect to our proxy model, I converted internal state to event for REE-side proxy.



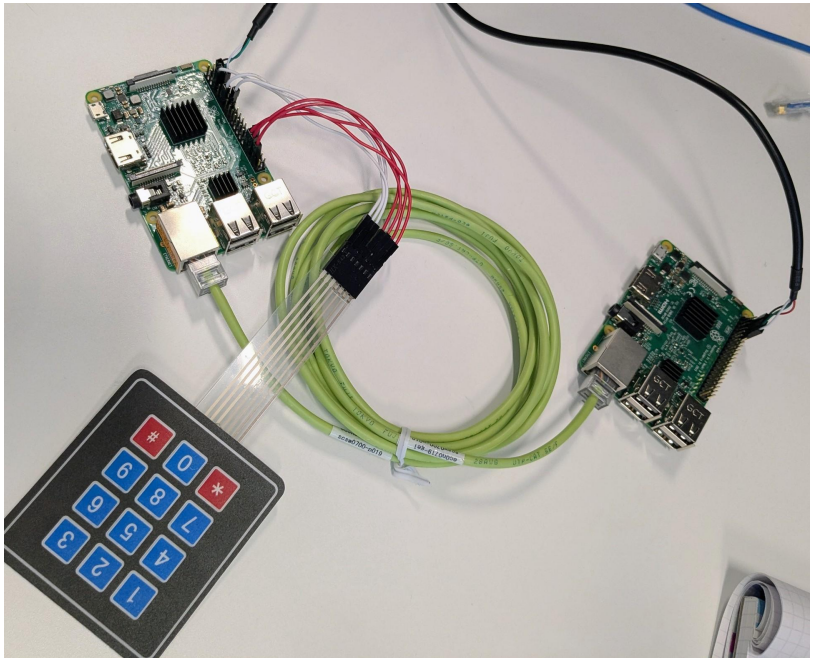

Implementation Problem



# Performance Evaluation Environment

Hardware:

- Raspberry Pi 3B V1.2 as a client/server
  - Cortex-A53 1.2GHz 4 core
  - connected with 100 Base-t Ethernet
  - TrustZone-A support
- keypad with GPIO

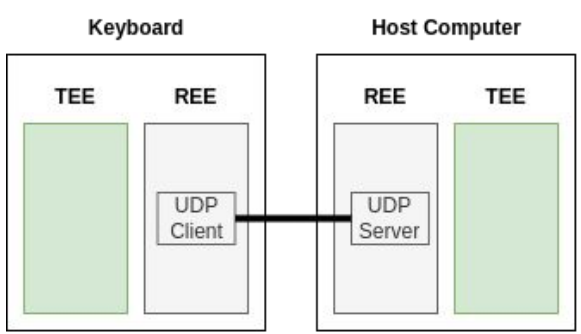



Software:

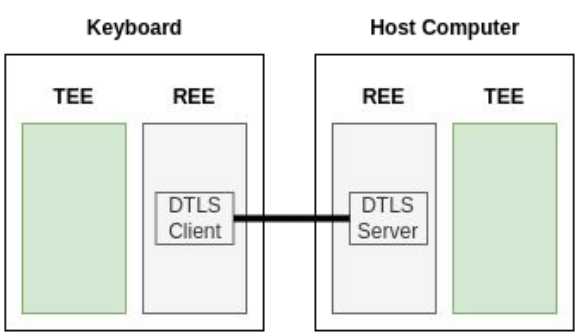
- TrustedFirmware-A v2.6
- OP-TEE OS 4.7.0
- Uboot 2021.10
- MbedTLS v3.6.0
- Linux v6.7.0-rc6-v8

# Performance Evaluation Workloads

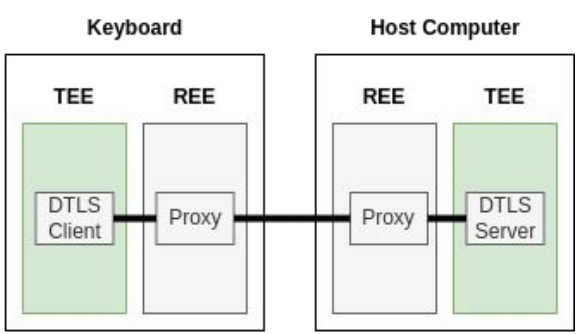
- UDP/DTLS throughput
- 3 cases:
  a. REE to REE UDP: max performance
  b. REE to REE DTLS: overhead of DTLS
  c. TEE to TEE DTLS: overhead of TEE




(a) REE to REE UDP Workload

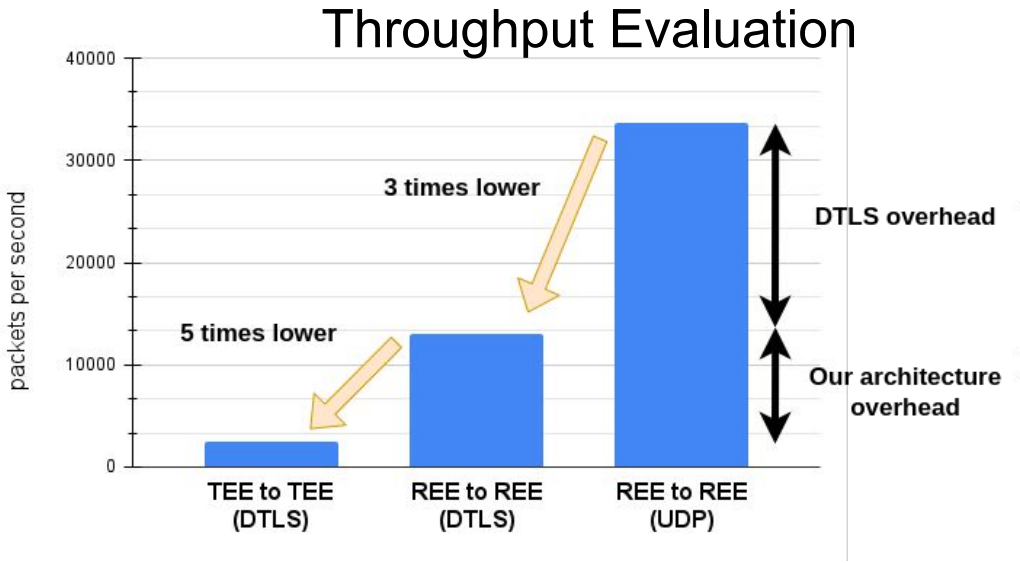


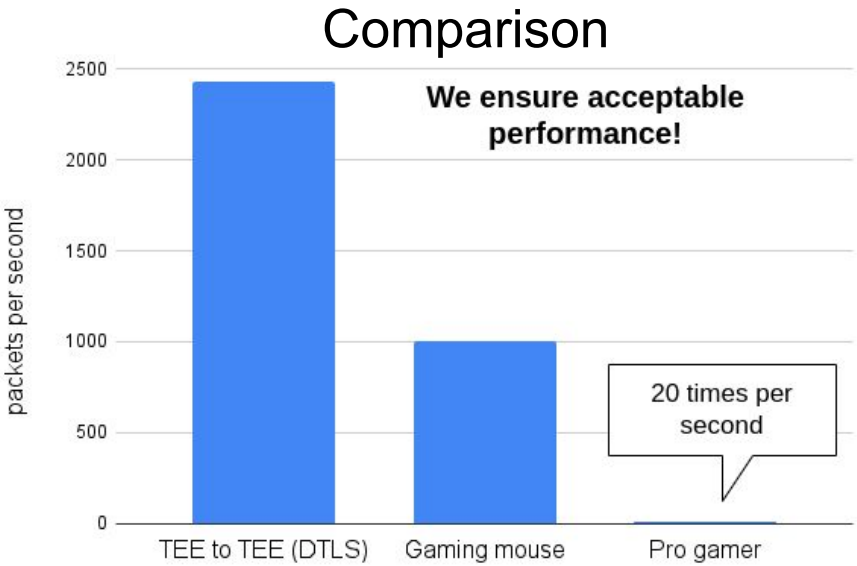

(b) REE to REE DTLS Workload




(c) TEE to TEE DTLS Workload

# Performance Evaluation Results (Throughput)




- Comparison
  - Pro gamer: 10 actions per second
  - Gaming mouse: 1000 updates per second

# Conclusion

- **Problem:** HID connection not secure, attacker can learn secrets or input keys
- We propose End-to-End trusted connection between host computer and Human Input Devices (HID)
- Our evaluation indicates our proposal provides acceptable throughput and latency for HID

Future Works:

- Implementation for TrustZone-M (Pi Pico 2)
- Implement attestation and firmware update mechanisms
- Use Ethernet over USB

# インターンについて

- **インターン経歴**
  - FFRI Internship 2023, 1 week
    - 開発経験や職場体験
  - GMO Pepabo Internship 2024 2 weeks
    - 開発系、実際のサービスに貢献
  - SECOM Lab Internship 2024, 2 weeks
    - 調査系、OSSの解析 （R&Dのような開発もある）
  - IIJ Lab Internship 2025, 2 months
    - 研究系、調査・実装・評価・論文記述