

Securing Embedded Systems with fTPM implemented as Trusted Application in TEE



FOSDEM 2024

Tymoteusz Burak





Tymoteusz Burak
Junior Embedded Systems Developer

-  tymoteusz.burak@3mdeb.com
-  [linkedin.com/in/tymoteusz-burak-a108252a0](https://www.linkedin.com/in/tymoteusz-burak-a108252a0)
- 8 months in 3mdeb
- Integration of functionalities and the creation of Operating Systems for embedded devices in Yocto
- Wrapping up my Bachelor's Degree in Automation and Robotics

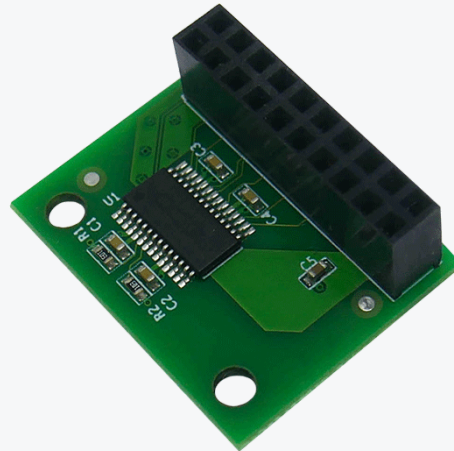


- coreboot licensed service providers since 2016 and leadership participants
- UEFI Adopters since 2018
- Yocto Participants and Embedded Linux experts since 2019
- Official consultants for Linux Foundation fwupd/LVFS project since 2020
- IBM OpenPOWER Foundation members since 2020

- What is TPM?
- What is fTPM?
- What is Arm TrustZone and how does it relate to fTPM?
- Arm TrustZone on different Cortex series
- Implementing fTPM in practice

What is TPM (Trusted Platform Module)?

*"A computer chip (microcontroller) that can securely store artifacts used to authenticate the platform (your PC or laptop). These artifacts can include passwords, certificates, or encryption keys. A TPM can also be used to store platform measurements that help ensure that the platform remains trustworthy."**



* quote from [Trusted Computing Group® - Trusted Platform Module \(TPM\) Summary](#).

Isolation from Host OS:

- Stores Secrets Safely: Encrypts and stores cryptographic keys separately from the host operating system, enhancing security.
- Protection Against Tampering: Secrets (like encryption keys) are not exposed to software vulnerabilities, protecting against unauthorized access and tampering.

System Integrity Verification:

- Storing Measurements in PCR Registers: During the boot process, the TPM measures components (like BIOS, bootloader, OS) and stores these measurements as hashes in PCR registers.
- Detects Changes: Any alteration in the boot process changes these measurements, enabling detection of unauthorized changes or tampering.

Secure Random Number Generation:

- **High-Quality Random Number Generation:** TPM includes a hardware-based random number generator (RNG) to produce cryptographically secure random numbers. These numbers are essential for creating unique encryption keys and for various cryptographic operations, ensuring that the cryptographic processes are robust against attacks.
- **Enhances Cryptographic Security:** By providing a source of entropy that is less predictable than software-based RNGs, the TPM's RNG strengthens the security of cryptographic operations as it's more difficult for attackers to predict or reproduce the cryptographic keys.

- Discrete TPM
- Integrated TPM
- Software TPM
- Firmware TPM

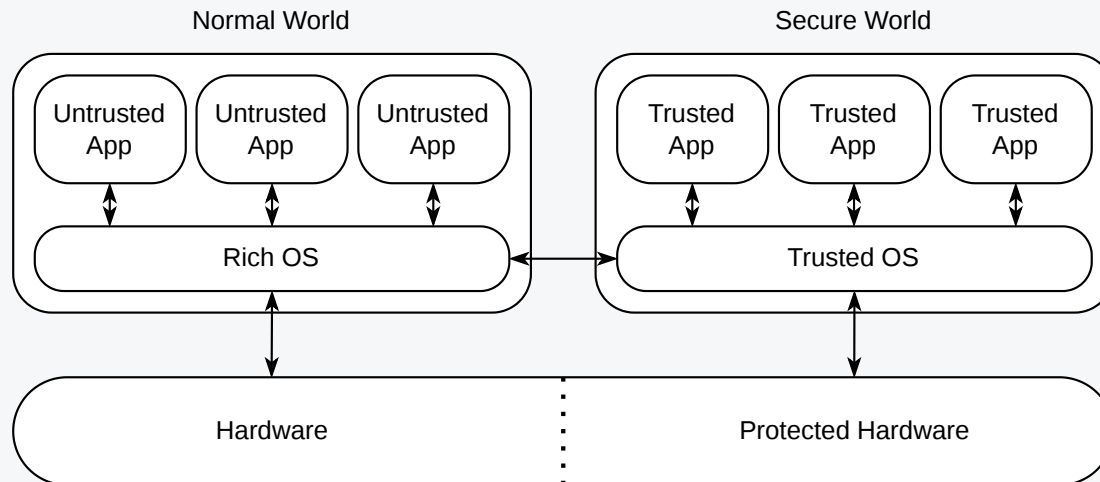
- Discrete TPM
 - Most secure
 - Separate physical chip installed on the motherboard
 - Operates independently from the main CPU
- Integrated TPM
- Software TPM
- Firmware TPM

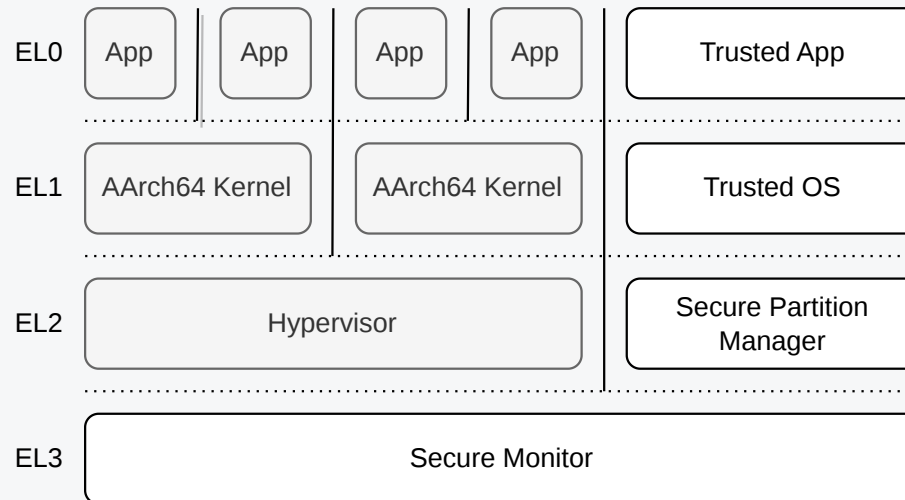
- Discrete TPM
 - Most secure
 - Separate physical chip installed on the motherboard
 - Operates independently from the main CPU
- Integrated TPM
 - Integrated into another chip that provides functions other than security
 - Economical and space-saving, reducing the need for additional components
 - Integration makes it less tamper-resistant
- Software TPM
- Firmware TPM

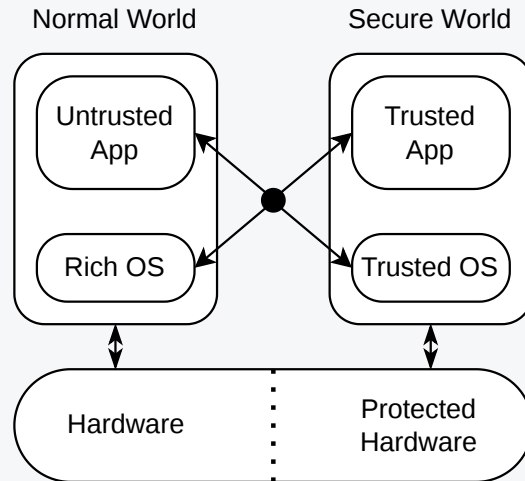
- Discrete TPM
 - Most secure
 - Separate physical chip installed on the motherboard
 - Operates independently from the main CPU
- Integrated TPM
 - Integrated into another chip that provides functions other than security
 - Economical and space-saving, reducing the need for additional components
 - Integration makes it less tamper-resistant
- Software TPM
 - Software emulation that runs in the user space
 - Least secure and susceptible to software attacks
 - Useful for testing and prototyping
- Firmware TPM

- Firmware TPM
 - Software implementation that runs in a Trusted Execution Environment (TEE)
 - Is separated from the rest of the programs that are running on the CPU
 - Cheap and can be implemented on already existing devices

"Confidential Computing is the protection of data in use by performing computation in a hardware-based, attested Trusted Execution Environment (TEE)."







The best protected systems have dedicated hardware security measures included from the beginning of their design process, starting with the specification for the processor core and the SoC infrastructure.

- No secure storage*
- No secure counter
- No secure clock
- No secure source of entropy*

- Coldboot
- Bus sniffing
- JTAG
- Worlds can't run in parallel
 - Running operations on fTPM freezes the normal OS

1. Build OP-TEE for your platform
2. Build fTPM as a TA for OP-TEE
3. Add userspace TPM/fTPM support

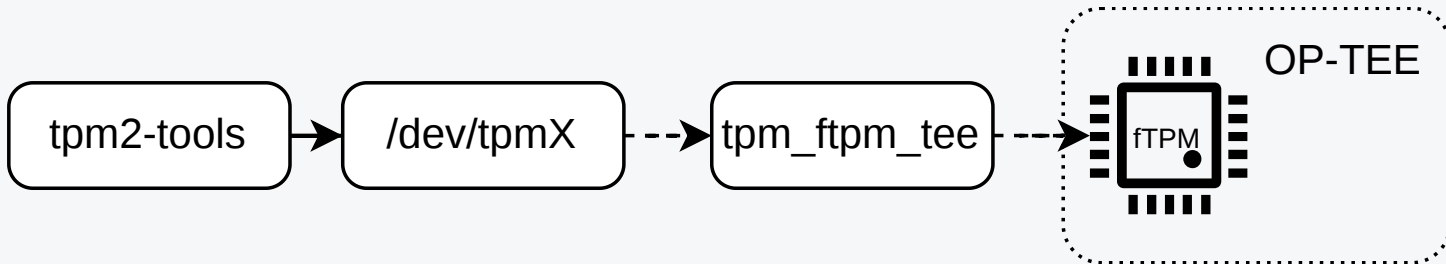
drivers/char/tpm/tpm_ftpm_tee.c

```
// SPDX-License-Identifier: GPL-2.0
/*
 * Copyright (C) Microsoft Corporation
 *
 * Implements a firmware TPM as described here:
 * https://www.microsoft.com/en-us/research/publication/ftpm-software-implementation-tpm-chip
 *
 * A reference implementation is available here:
 * https://github.com/microsoft/ms-tpm-20-ref/tree/master/Samples/ARM32-FirmwareTPM/optee\_ta/
 */
(...)
```

TPM



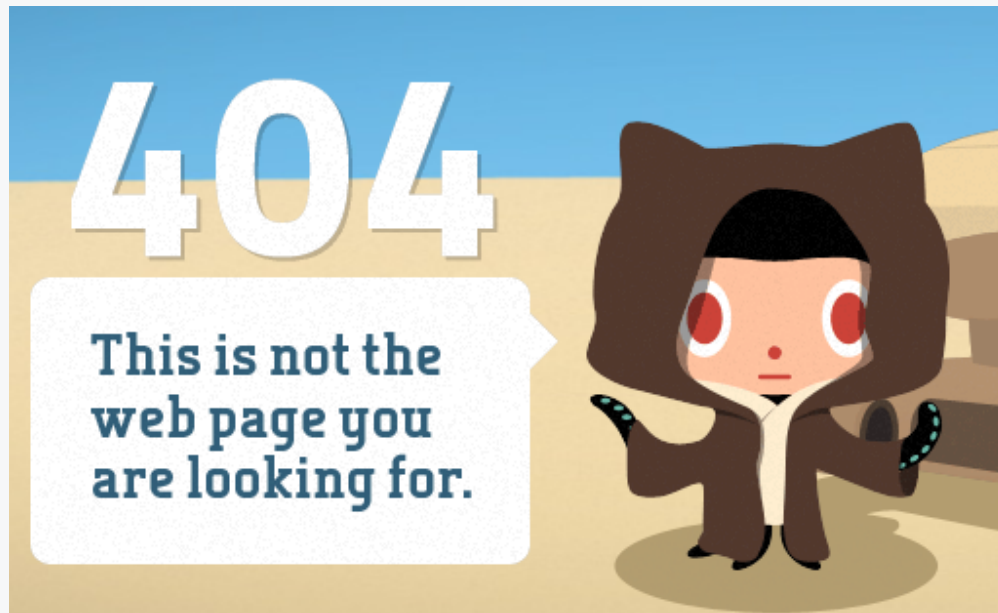
fTPM



4. Clone the OpTEE OS source code

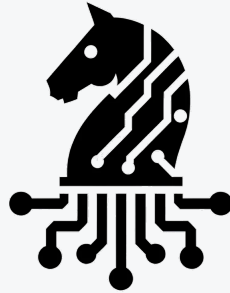
If you do not already have a version of the OP-TEE OS repo cloned on your machine you may run:

```
cd ~  
git clone https://github.com/ms-iot/ms-iot-optee_os.git
```







meta-arm - optee-tpm_git.bb

```
SUMMARY = "OPTEE fTPM Microsoft TA"  
DESCRIPTION = "TCG reference implementation of the TPM 2.0 Specification."  
HOMEPAGE = "https://github.com/microsoft/ms-tpm-20-ref/"  
  
COMPATIBLE_MACHINE ?= "invalid"  
COMPATIBLE_MACHINE:qemuarm64 = "qemuarm64"  
COMPATIBLE_MACHINE:qemuarm64-secureboot = "qemuarm64"  
COMPATIBLE_MACHINE:qemu-generic-arm64 = "qemu-generic-arm64"  
COMPATIBLE_MACHINE:qemuarm-secureboot = "qemuarm"  
  
(...)
```



Zarhus OS

-  [linkedin.com/company/3mdeb](https://www.linkedin.com/company/3mdeb)
-  contact@3mdeb.com
-  [facebook.com/3mdeb](https://www.facebook.com/3mdeb)
-  [@3mdeb_com](https://twitter.com/_@3mdeb_com)
- <https://3mdeb.com>
- [Book a call](#)
- [Sign up for the newsletter](#)

Feel free to contact us if you believe we can help you in any way. We are always open to cooperate and discuss

- Will Arthur, David Challener, and Kenneth Goldman. *A Practical Guide to TPM 2.0: Using the Trusted Platform Module in the New Age of Security*. Apress Berkeley, CA, 2015. DOI: 10.1007/978-1-4302-6584-9.
- Sandro Pinto and Nuno Santos. 2019. *Demystifying Arm TrustZone: A Comprehensive Survey*. ACM Comput. Surv. 51, 6, Article 130 (January 2019), 36 pages. <https://doi.org/10.1145/3291047>
- [fTPM: A Software-only Implementation of a TPM Chip](#)

Himanshu Raj, ContainerX; Stefan Saroiu, Alec Wolman, Ronald Aigner, Jeremiah Cox, Paul England, Chris Fenner, Kinshuman Kinshumann, Jork Loeser, Dennis Mattoon, Magnus Nystrom, David Robinson, Rob Spiger, Stefan Thom, and David Wooten, Microsoft

[USENIX Security '16 presentation by Stefan Saroiu](#)
- [Trusted Computing Group - TPM 2.0 A Brief Introduction](#)
- [TEEs are not Silver Bullets - David Cerdeira](#)
- [OP-TEE Documentation](#)
- [Introduction to Trusted Execution Environment and ARM's TrustZone](#)
- [Develop Secure Cortex-M Applications with Trustzone - Kristoffer Martinsson](#)
- [jbech-linaro/manifest](#) and [jbech-linaro/docker_optee](#)

Q&A