Main Site › Articles

Knowledge article

Understanding Open Source Trusted Execution Environment - OP-TEE

Created 1 month ago Active 1 month ago Last edited 1 month ago Viewed 15 times 1 min read

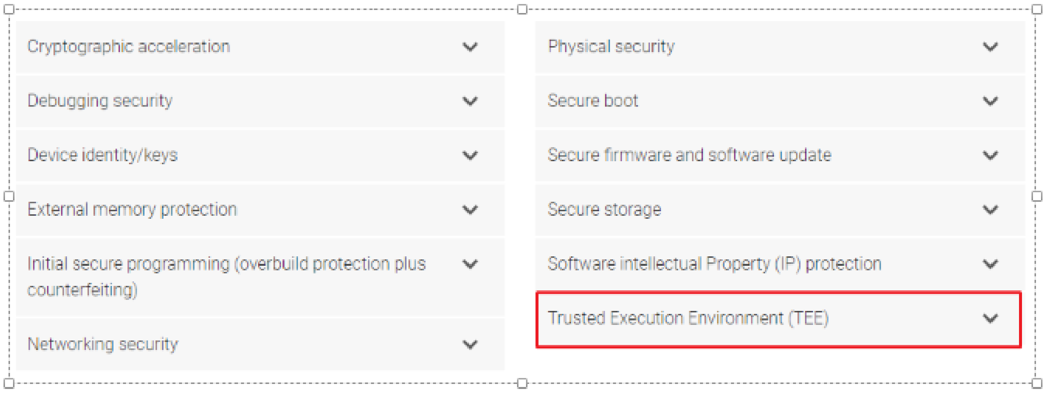
security-optee

Share article ••• Edit article

2

Security for any application in embedded product should answer questions like
Q. What is being protected? (Asset)
Q. Who or what are we protecting against? (Threat and threat probability)
Q. What is the attack surface? (Exposure points and threat probability)

Based on the answers we get we would have various possibilities/enablers available across processors families to enable those aspects of security. Refer security enablers by TI



- Cryptographic acceleration
- Debugging security
- Device identity/keys
- External memory protection
- Initial secure programming (overbuild protection plus counterfeiting)
- Networking security
- Physical security
- Secure boot
- Secure firmware and software update
- Secure storage
- Software intellectual Property (IP) protection
- Trusted Execution Environment (TEE)

Before we begin understanding of OP-TEE, lets understand the TrustZone

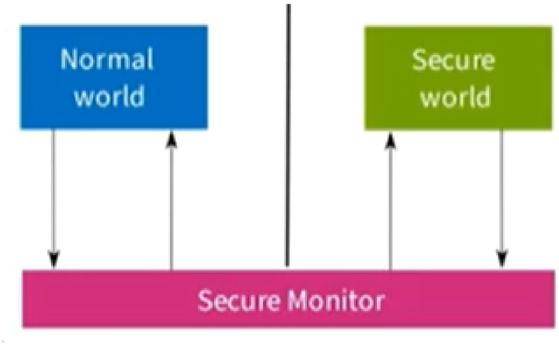
1. It’s a ARM hardware feature, its available on almost all modern ARM V7 SOC
2. It means processor has ability to switch between normal and secure world
3. Normal world, running i.e. Linux

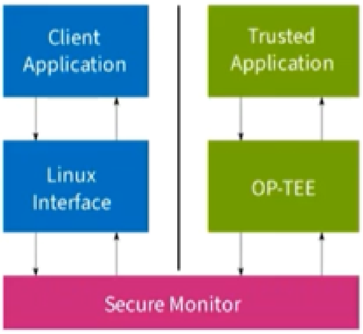
4. Secure world, running i.e. OP-TEE (Its execution environment which isn't complete operating system - its OP-TEE for us)
5. Normal world communicate with secure world using secure monitor
6. Secure monitor will ensure that memory address are translated and no data leaks between two diff. worlds
7. General idea is secure world not accessible from normal world
8. Secure world can do secure operations like cryptographic operations and this isn;t accessible to normal world

What is OP-TEE

1. OP-TEE - Open Source Trust Environment Execution
2. Its implementation of Global Platform Trust Execution Environment specification
3. Idea is to write your trusted application independent of the specific trusted execution environment you want run your application on
4. Not mandate to use OPTEE but any environment that complies with TEE specification
5. BSD 2/3 - clause licensed

> - Include Cryptography libs- like libtomcrypt , mbedTLS 6. OP-TEE provides execution environment, it's not a OS
> - provides no scheduling or preemption
> - Processor runs with OP-TEE it will execute specific task and then passes control to normal world for execution there
> - We can decide rate of application run in secure run environment 7. General idea is to use small part of application and put it in secure application zone, like -Big part is still in Linux-REE -Application is split in two REE(That’s linux) and TEE
> - On Linux side we can use LibTee that implements interface for TA communication





OP-TEE Features

1. Using Replay Protected Memory Blocks (eMMC/NVME feature) for rollback protected storage
2. Drivers for common DDR access firewalls (TZC380, TZC400)
3. Upstream kernel driver maintained by OP-TEE maintainers
4. Platform Support for: i.MX, Layerscape, STM32MP1, qemu, hikey, raspberry pi 3, rockchip and TI AMxx

OP-TEE Use cases

1. TPM (PCR, Sealing, Attestation)
2. PKCS#II (i.e. Signing, Device Authentication)
3. Trusted Keys (Linux Keyring Sealing, under discussion)
4. Payment verification?
5. Content decryption (DRM)?
6. License Management?

References

TI-REFERENCE
OP-TEE is Ready

edited Jan 9 at 4:50

created Jan 8 at 4:43
mahesh-gaikwad
753 ● 1 ● 11

Good Read Mahesh, thank you – Shantanu Jan 9 at 4:43