2023-04-08

Created: 2023-04-29 22:45:29

Questions

- 2020 Demystifying TEE Client API `(230406.Papers.OP-TEE.txt)`
  - `[11:30]` Paper: Demystifying ARM TrustZone TEE Client API using OP-TEE `(Papers.OP-TEE.0406.md)`
- Virtualization `(230327.OP-TEE.txt)`
  - `[23:11]` The following article are from the website https://optee.readthedocs.io/en/latest/architecture/virtualization.html `(OP-TEE.0327.md)`
  - `[23:15]` Does the hypervisor relied on OP-TEE run in normal world or secure world?`(OP-TEE.0327.md)`
  - `[23:19]` Does a TA run in a VM described in the article?`(OP-TEE.0327.md)`

Chats

- OP-TEE.0327.md
  - `[11:00]` Can OP-TEE be ported in Qualcomm devices?`(OP-TEE.0327.md)`
  - `[11:02]` Who did port OP-TEE in Qualcomm devices successfully?`(OP-TEE.0327.md)`
  - `[23:25]` What is SPMC? `(OP-TEE.0327.md)`
  - `[23:27]` Is SPMC of TrustZone? `(OP-TEE.0327.md)`
  - `[23:47]` What is FF-A? `(OP-TEE.0327.md)`
- OP-TEE.History.0407.md
  - `[11:42]` Please list history of Linaro OP-TEE with years. `(OP-TEE.History.0407.md)`
  - `[11:43]` Please list history of Linaro OP-TEE with years after 2015.`(OP-TEE.History.0407.md)`
  - `[11:45]` Please list history of Qualcomm QTEE with years after 2017.`(OP-TEE.History.0407.md)`
- TrustZone.0329.md
  - `[23:32]` Is Secure Partition of TrustZone? `(TrustZone.0329.md)`
  - `[23:33]` Is SPMC of TrustZone? `(TrustZone.0329.md)`

Q & A

2020 Demystifying TEE Client API `(230406.Papers.OP-TEE.txt)`

Question: `[11:30]` Paper: Demystifying ARM TrustZone TEE Client API using OP-TEE `(Papers.OP-TEE.0406.md)`

Answer:

Title: Demystifying ARM TrustZone TEE Client API using OP-TEE

Abstract: With the increased use of mobile devices for various purposes, such as payment systems, the need for secure storage of critical data has become more important. Trusted Execution Environment (TEE) technology has emerged as a solution to provide a secure execution environment for sensitive data. ARM TrustZone technology, primarily used in mobile devices, divides a physical processor into the Normal World and Secure World, ensuring a safer execution environment. While various manufacturers have adopted TrustZone technology, existing commercial TEEs have limitations in conducting security research using TrustZone.

This paper introduces OP-TEE, an open-source project for implementing ARM TrustZone technology, and the TEE Client API that enables communication with Trusted Applications within the TrustZone Secure World. To demystify the TEE Client API, the authors also implemented a simple trusted application for communication between the Normal World and Secure World in the OP-TEE OS using the QEMU emulator. This research aims to make the TEE Client API more accessible and easier to understand for researchers and developers working with ARM TrustZone technology.

Virtualization `(230327.OP-TEE.txt)`

Question: `[23:11]` The following article are from the website https://optee.readthedocs.io/en/latest/architecture/virtualization.html `(OP-TEE.0327.md)`

Answer:

The following is a summary of OP-TEE's experimental virtualization support, as described in the article:

- One OP-TEE instance can run TAs from multiple virtual machines, isolating all VM-related states to prevent one VM from affecting another.
- OP-TEE relies on a hypervisor (e.g., Xen) to determine which VM is calling it, and the hypervisor must inform OP-TEE about the creation and destruction of VMs.
- OP-TEE cannot translate intermediate physical addresses (IPAs) to physical addresses (PAs), so the hypervisor is responsible for this translation.
- Virtualization support is enabled with the CFG_VIRTUALIZATION configuration option, and the maximum number of supported VMs is controlled by CFG_VIRT_GUEST_COUNT.
- Hypervisors must be aware of OP-TEE, handle SMCs from virtual guests, perform IPA<->PA address translation, and manage memory pages shared between VMs and OP-TEE.
- Limitations of the experimental virtualization support include platform support (only Armv8 is supported and tested on QEMU-V8), static VM guest count and memory allocation, sharing hardware resources and PTAs, compatibility with "normal" mode, and the experimental state of the technology.
- The implementation details of OP-TEE virtualization support involve splitting OP-TEE into "nexus" (core part) and TEE (handles requests and runs TAs). There is one nexus instance and multiple TEE instances, one for each registered VM. Banked memory sections are used for TEE/VM instances, and the nexus and TEE have separate heaps.

This summary provides an overview of OP-TEE's experimental virtualization support, but it is essential to refer to the original documentation for a complete understanding of the technology, its requirements, and its limitations.

Question: `[23:15]` Does the hypervisor relied on OP-TEE run in normal world or secure world?`(OP-TEE.0327.md)`

Answer:

The hypervisor that OP-TEE relies on runs in the normal world. It is responsible for managing multiple virtual machines and handling the communication between the normal world and OP-TEE in the secure world. The hypervisor must be aware of OP-TEE, handle SMCs from virtual guests, perform IPA <-> PA address translation, and manage memory pages shared between VMs and OP-TEE.

Question: `[23:19]` Does a TA run in a VM described in the article?`(OP-TEE.0327.md)`

Answer:

No, a Trusted Application (TA) does not run inside a Virtual Machine (VM) as described in the article. Instead, the TA runs in the secure world within the OP-TEE environment. The hypervisor manages the VMs in the normal world and is responsible for handling the communication between the normal world and OP-TEE in the secure world. When OP-TEE has virtualization support enabled, it allows multiple VMs to utilize a single OP-TEE instance for running TAs while maintaining isolation between the VM-related states.

OP-TEE.0327.md

Question: `[11:00]` Can OP-TEE be ported in Qualcomm devices?`(OP-TEE.0327.md)`

Answer:

Yes, OP-TEE can be ported to Qualcomm devices, though it may require some additional work to make it compatible with the specific hardware. Since Qualcomm has its own Trusted Execution Environment (TEE) called QSEE (Qualcomm Secure Execution Environment), you would need to replace or adapt it to work with OP-TEE.

Porting OP-TEE to a Qualcomm device involves several steps:

1. Acquiring documentation and source code for the target device, including information on the processor, memory layout, peripherals, and TrustZone implementation.
2. Adapting the OP-TEE platform layer to support the specific hardware features of the target device, including modifications to the memory layout, interrupt handling, and device drivers.
3. Integrating OP-TEE with the bootloader and normal-world OS (e.g., Android), ensuring the correct configuration and loading of OP-TEE during the boot process.
4. Testing and debugging the OP-TEE implementation on the target device, ensuring the correct functioning of the secure world, normal world, and interactions between the two.

Keep in mind that porting OP-TEE to a Qualcomm device may require deep technical expertise and familiarity with the target hardware, OP-TEE internals, and the TrustZone architecture. Additionally, access to detailed documentation, tools, and support from Qualcomm may be necessary for a successful port.

Question: `[11:02]` Who did port OP-TEE in Qualcomm devices successfully? `(OP–TEE.0327.md)`

Answer:

There isn't a comprehensive list of all individuals or organizations that have successfully ported OP-TEE to Qualcomm devices. However, Linaro, a collaborative engineering organization that focuses on developing open-source software for ARM hardware, has worked on porting OP-TEE to a variety of platforms, including some Qualcomm devices.

Linaro has contributed significantly to the development of OP-TEE, and they have expertise in porting OP-TEE to different hardware platforms. Some of their work involves using development boards, such as the DragonBoard 410c, which is based on the Qualcomm Snapdragon 410 chipset. Linaro has also provided some level of support for other Qualcomm platforms, though the availability of support and documentation may vary depending on the specific device.

It is worth noting that individual developers and companies may also have ported OP-TEE to Qualcomm devices for their own projects, but such information may not be publicly available due to proprietary reasons or non-disclosure agreements.

Question: `[23:25]` What is SPMC? `(OP–TEE.0327.md)`

Answer:

The SPMC (Secure Partition Manager Core) is a component within ARM's virtualization architecture for ARMv8.4-A systems. The SPMC manages secure partitions (SPs), which are isolated execution environments that run within the secure world of a virtualized ARM system. Secure partitions are similar to Trusted Applications (TAs) in OP-TEE, as they provide a secure environment for running trusted code.

The SPMC is responsible for:

1. Lifecycle management of secure partitions, including their creation, initialization, and termination.
2. Managing inter-partition communication and resource sharing between secure partitions.
3. Managing communication between the normal world (Non-secure world) and secure world, including handling requests and responses from the normal world.

The SPMC ensures that secure partitions can execute in a safe and isolated environment, providing a higher level of security and separation for trusted applications in a virtualized ARM system.

Question: `[23:27]` Is SPMC of TrustZone? `(OP–TEE.0327.md)`

Answer:

The SPMC (Secure Partition Manager Core) is not a component of TrustZone specifically. TrustZone is a hardware-based security feature implemented in ARM processors, which creates a secure world and a normal world to separate and protect sensitive data and code execution.

SPMC, on the other hand, is a part of ARM's virtualization architecture for ARMv8.4-A systems, which is designed to manage secure partitions (SPs) within the secure world. While SPMC is not a direct component of TrustZone, it relies on the security features provided by TrustZone, as secure partitions execute within the secure world created by TrustZone.

In summary, TrustZone provides the foundation for secure execution environments, while SPMC manages and orchestrates secure partitions within that environment in the context of virtualized ARM systems.

Question: `[23:47]` What is FF-A? `(OP-TEE.0327.md)`

Answer:

FF-A, or Firmware Framework for Arm, is a specification that defines a standard interface for communication between various software components running on Arm-based systems. This interface enables secure and efficient communication between different execution environments, such as:

- Normal world operating systems (e.g., Linux, Android)
- Secure world trusted execution environments (e.g., OP-TEE, TrustZone)
- Hypervisors
- Secure partitions managed by a Secure Partition Manager (SPM)

FF-A is designed to provide a common interface across different Arm platforms, making it easier for developers to create software that works consistently across various Arm-based devices. The FF-A specification includes a standardized set of Service Function Identifiers (SFIDs) for commonly used services, as well as a mechanism for vendor-specific extensions.

In the context of virtualization and TEEs like OP-TEE, FF-A can be used to facilitate communication between normal world applications, the hypervisor, and secure partitions, improving security and simplifying software development.

OP-TEE.History.0407.md

Question: `[11:42]` Please list history of Linaro OP-TEE with years. `(OP-TEE.History.0407.md)`

Answer:

Linaro is a collaborative engineering organization that maintains and develops the open-source OP-TEE (Open Portable Trusted Execution Environment) project. Here's a brief history of the Linaro OP-TEE project with relevant years:

1. 2013: OP-TEE project initiation - STMicroelectronics initiates the OP-TEE project, aiming to create an open-source TEE for ARM TrustZone-enabled platforms.

2. 2014: Linaro Security Working Group involvement - The Linaro Security Working Group begins working on the OP-TEE project, collaborating with STMicroelectronics and other stakeholders.

3. 2015: Linaro takes over maintenance and development - Linaro assumes responsibility for the maintenance and development of the OP-TEE project, providing regular releases, bug fixes, and ongoing support.

Since 2015, Linaro has continued to actively maintain and develop the OP-TEE project, making it a valuable resource for implementing Trusted Execution Environments on ARM TrustZone-enabled platforms.

Question: `[11:43]` Please list history of Linaro OP-TEE with years after 2015. `(OP-TEE.History.0407.md)`

Answer:

Since taking over maintenance and development of the OP-TEE project in 2015, Linaro has continued to actively maintain and develop the project. While there may not be specific landmark events, here are some notable developments in the project's history after 2015:

1. 2016: OP-TEE v2.2.0 release - This release includes various feature additions, bug fixes, and improvements to the project.

2. 2017: OP-TEE v3.0.0 release - The project moves to a new major version (v3.x.x), indicating significant changes and

improvements. This release contains updates to the GlobalPlatform TEE Internal Core API specification.

3. 2018: OP-TEE v3.3.0 release - This release features support for ARMv8-A Scalable Vector Extension (SVE) and updates to the GlobalPlatform API specifications.
4. 2019: OP-TEE v3.6.0 release - The release includes important bug fixes, improvements, and support for virtualization using ARMv8-A Virtualization Host Extensions (VHE).
5. 2020: OP-TEE v3.10.0 release - This release brings several feature additions, bug fixes, and improvements. Notably, it adds support for ARMv8.4-A architecture.

Throughout the years, Linaro has provided regular updates to the OP-TEE project, improving its stability, security, and compatibility with a variety of platforms. Although the list above highlights some significant releases, it's important to note that Linaro releases multiple versions each year, which include incremental improvements, bug fixes, and new features to the OP-TEE project. The detailed release notes can be found on the project's GitHub repository: https://github.com/OP-TEE/optee_os/releases

---

Question: `[11:45]` Please list history of Qualcomm QTEE with years after 2017. `(OP-TEE.History.0407.md)`

Answer:

After further investigation, it appears that public information on specific Qualcomm QTEE developments and updates after 2017 is limited. However, Qualcomm has continuously worked to improve security features in their Snapdragon SoCs, including enhancements to the Qualcomm Trusted Execution Environment (QTEE) and the Secure Processing Unit (SPU) introduced in 2017.

It's important to note that as a proprietary technology, the detailed history and updates for QTEE may not be as readily available as they are for open-source projects like Linaro's OP-TEE.

Nonetheless, Qualcomm's ongoing commitment to security can be observed through regular releases of new Snapdragon SoCs, which integrate improved security features and leverage ARM TrustZone technology.

TrustZone.0329.md

Question: `[23:32]` Is Secure Partition of TrustZone? `(TrustZone.0329.md)`

Answer:

A Secure Partition (SP) is a component within the ARM TrustZone architecture. TrustZone provides a secure execution environment by partitioning the system into two separate worlds: the Secure World and the Normal World. Secure Partitions are entities that run in the Secure World, which is specifically designed to protect sensitive code and data from unauthorized access or tampering.

Secure Partitions host trusted applications (TAs) or other security-sensitive services, and they are isolated from each other and the Normal World to maintain a high level of security. Each Secure Partition runs in its own isolated environment with limited access to system resources, ensuring that even if one partition is compromised, it cannot impact the others.

In summary, Secure Partitions are an integral part of the ARM TrustZone architecture, providing a secure environment for running trusted applications and services.

Question: `[23:33]` Is SPMC of TrustZone? `(TrustZone.0329.md)`

Answer:

The Secure Partition Manager (SPM) is a component within the ARM TrustZone architecture. SPMC stands for Secure Partition Manager Core, which is the core functionality of the SPM. The SPMC is responsible for managing Secure Partitions (SPs) within the TrustZone Secure World.

The SPMC handles tasks such as creating, configuring, and managing Secure Partitions, as well as inter-partition communication and resource management. It also provides a secure environment for trusted applications (TAs) to run within the Secure World, ensuring that sensitive code and data are protected from unauthorized access or tampering.

In summary, the SPMC is a key component of the ARM TrustZone architecture, responsible for managing Secure Partitions and providing a secure environment for trusted applications to run.