230330.OP-TEE.j721e

Created: 2023-04-29 22:44:36

Questions

- Basic Concepts
  - https://software-dl.ti.com/jacinto7/esd/processor-sdk-linux-rt-jacinto7/08_04_00_03/exports/docs/linux/Foundational_Components_OPTEE.html?highlight=trusted `(OP-TEE.j721e.0330.md)`
  - What is GP? `(OP-TEE.j721e.0330.md)`
  - What is RPMB? `(OP-TEE.j721e.0330.md)`
  - What is KEK? `(OP-TEE.j721e.0330.md)`
  - Is a KEK unique for each device? `(OP-TEE.j721e.0330.md)`

Q & A

Basic Concepts

Question: https://software-dl.ti.com/jacinto7/esd/processor-sdk-linux-rt-jacinto7/08_04_00_03/exports/docs/linux/Foundational_Components_OPTEE.html?highlight=trusted `(OP-TEE.j721e.0330.md)`

Answer:

Based on the information provided, the instructions explain how to build and integrate OP-TEE with the Jacinto 7 Processor for both GP and HS devices. Here’s a summary of the steps:

1. Get the OP-TEE Source Code: Clone the OP-TEE source code from GitHub.
2. Get the Security Dev Tool: Clone the TI Security Development Tools repository and set the TI_SECURE_DEV_PKG environment variable.
3. Build OP-TEE: Build the OP-TEE binary for the k3-j721e platform using the provided commands.
4. Sign the image for HS: If you are building for an HS device, sign the tee-pager_v2.bin using the provided script.
5. Enable RPMB support and secure storage: To enable RPMB support along with secure storage, additional flags need to be passed to the build instructions. You can enable the hybrid mode of RPMB along with REE_FS by following the provided example.
6. Get and build the OP-TEE Client with RPMB support Clone the OP-TEE client source code and build it with the appropriate flags. Then, update the optee-client binary and libraries on your SD card with the generated ones in the out/export/usr folder.
7. Build U-Boot with OP-TEE OS: Follow the instructions to build U-Boot with the TEE parameter pointing to the newly built tee-pager_v2.bin.
8. Integrate binary output into U-Boot: Re-build U-Boot with A72/A53 instructions found under “Build U-Boot,” but with the TEE parameter pointing to the newly built tee-pager_v2.bin.

Remember that tee-pager_v2.bin may be referred to as bl32.bin in other documentation.

For additional information, you can refer to the official OP-TEE documentation at https://optee.readthedocs.io/en/latest/.

Please note that this is a summary of the instructions provided, and you should follow the detailed steps in the original text to build and integrate OP-TEE successfully.

Question: What is GP? `(OP-TEE.j721e.0330.md)`

Answer:

In the context of the instructions provided, “GP” refers to “General Purpose” devices. When discussing Texas Instruments (TI) processors, the term is used to differentiate between two types of devices: General Purpose (GP) and High Security (HS).

General Purpose (GP) devices are designed for a broad range of applications and use cases. They typically do not include the same level of security features as High Security (HS) devices.

High Security (HS) devices, on the other hand, are designed with additional security features and are intended for use in applications that require a higher level of security, such as automotive systems, industrial control systems, or other safety-critical applications.

When building OP-TEE for a Jacinto 7 Processor, you may need to follow different steps depending on whether you are targeting a General Purpose (GP) or High Security (HS) device. For example, the instructions provided mention signing the tee-pager_v2.bin when building for an HS device, which is not required for a GP device.

Question: What is RPMB? `(OP-TEE.j721e.0330.md)`

Answer:

RPMB stands for Replay Protected Memory Block. It is a security feature provided in eMMC (embedded Multi-Media Controller) and UFS (Universal Flash Storage) devices. RPMB is designed to offer a secure, tamper-resistant storage area for sensitive data and to protect against replay attacks.

In the context of OP-TEE and secure storage, RPMB is used to store sensitive information in a protected environment, ensuring data confidentiality and integrity. It can be used in combination with other secure storage options, such as REE_FS (Rich Execution Environment File System), to create a hybrid storage solution. The hybrid mode uses the Secure Storage directory in the RPMB partition to check for the integrity of the data present in it, providing an added layer of security.

When building OP-TEE with RPMB support, you need to pass additional flags to the build instructions, as mentioned in the provided steps. This enables the use of real eMMC storage instead of the default virtual eMMC storage.

Question: What is KEK? `(OP-TEE.j721e.0330.md)`

Answer:

KEK stands for Key Encryption Key. In cryptography and secure systems, a KEK is used to securely transmit and store encryption keys. The main purpose of a KEK is to protect the actual encryption keys (also known as data encryption keys or DEKs) that are used to encrypt and decrypt sensitive data.

The KEK is used to encrypt the DEKs before they are stored or transmitted, and only authorized systems with the correct KEK can decrypt the DEKs to access the actual encryption keys. This adds an additional layer of security, ensuring that even if the encrypted DEKs are intercepted or compromised, an attacker would still need the KEK to decrypt and use them.

In the context of TI SoCs (System-on-Chip) with High Security (HS) configuration, a KEK is embedded in the devices and programmed across OP-TEE instances. Each HS device has its own unique signing key, which is different from other HS devices. This further enhances the security of the overall system, making it more difficult for attackers to compromise the sensitive data stored within.

Question: Is a KEK unique for each device? `(OP-TEE.j721e.0330.md)`

Answer:

In the context of Texas Instruments (TI) System-on-Chip (SoC) devices with High Security (HS) configuration, each device does have a unique KEK (Key Encryption Key). This means that each HS device has its own signing key, which is different from other HS devices. This approach enhances the security of the overall system, making it more difficult for attackers to compromise the sensitive data stored within.

However, it’s essential to understand that the uniqueness of a KEK depends on the specific implementation and the security requirements of the system. In some cases, KEKs may be shared among multiple devices or systems to enable secure communication between them.

In summary, while the KEKs in TI SoCs with HS configuration are unique for each device, the uniqueness of KEKs in other systems may vary depending on the specific implementation and security requirements.