Ethan.Zhang · 2019年11月18日

# TrustZone和PSA之间是什么关系？

Arm Cortex-M PSA IoT TrustZone

熟悉Arm的朋友基本都听说过TrustZone和PSA，但是很多不太了解两者之间是什么关系。TrustZone是Arm架构的安全扩展，是系统级的安全方案，已经被业内广泛的应用，也有两篇文章介绍过TrustZone：

- Cortex-M和Cortex-A的TrustZone差异
- Arm的TrustZone，CryptoCell，以及CryptoIsland到底什么关系

PSA是Arm在2017年推出的平台安全架构，主要目的是实现成本可控，易于实施，低风险的物联网安全基础平台。PSA的文档和资料非常多，并且不再同一个网站，我们也对PSA的资料进行了汇总方便大家查看：

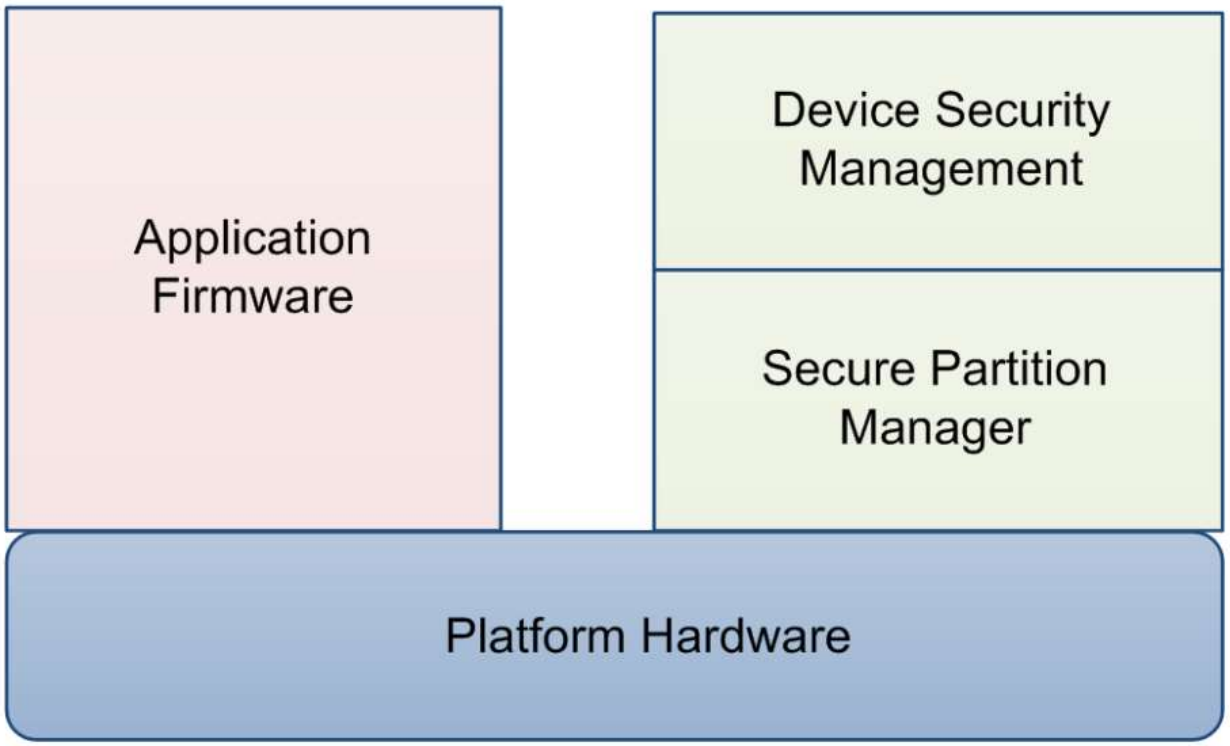
- PSA资料汇总

为了让大家方便理解两者之间的关系，我们以PSA level 1 SoC的安全要求为例来介绍，对于Level 1来说安全要求非常简单，只有4个要求分别是：

| ID | Requirement | Response | | |
|---|---|---|---|---|
| | | Yes | Part. | N/A |
| C1.1 | The chip has a hardware mechanism to isolate the Secure Processing Environment (SPE) and related assets from the Non-Secure Processing Environment. | | | |
| | *(Describe how isolation is implemented, typically through TrustZone on Cortex-v8M or dual cores on Cortex-v7M)*<br>*Example of response for Yes: The Cortex-M33 (ARMv8-M architecture) supports TrustZone. The Secure Processing Environment is executed in the Secure mode.* | | | |
| C1.2 | The chip provides trusted boot support, initiated from immutable code.<br>*NB: Immutable code can be for instance ROM, or EEPROM or FLASH memory that is locked before device delivery.* | | | |
| | *(Describe which cryptographic functions and key sizes are used for trusted boot, and how cryptography is implemented, such as hardware cryptographic accelerator or software in immutable code. Also describe how locking is performed if boot code is stored in mutable memory such as EEPROM or FLASH)*<br>*Example of response for Part: The Boot ROM runs the Bootloader in secure mode but without prior validation. The Bootloader authenticates the SPE image by hash (SHA-256) and digital signature (RSA-2048) validation. Public key is built into the bootloader image. Metadata of the image is delivered together with the image itself in a header and trailer section. In case of successful authentication, bootloader passes execution to the secure image.* | | | |
| C1.3 | The chip supports security lifecycle, i.e. protect a lifecycle state for the device and enforce transition rules between states.<br>The supported lifecycle states should include at least Device assembly and Test, Factory provisioning, Provisioned and a Debug mode.<br>*NB: This requirement is not mandatory for the first products that will be evaluated in 2019.* | | | |
| | *(Describe supported lifecycle states and transition rules)*<br>*Example of response for Yes: The chip supports security lifecycle as defined in [PSA-SM], §E - Generic PSA security lifecycle.* | | | |
| C1.4 | The chip supports the secure storage of following keys:<br>• Hardware Unique Key (HUK), with at least with 256-bits of entropy, used for deriving other per device secrets<br>• RoT Public Key (RoTPK), used for authenticating the first stage of SPE code during trusted boot<br>• Unique attestation key (see requirement below).<br>These keys may be injected during initial manufacturing of the silicon or during the final manufacturing of a product or also be derived from a Physically Unique Function (PUF). | | | |

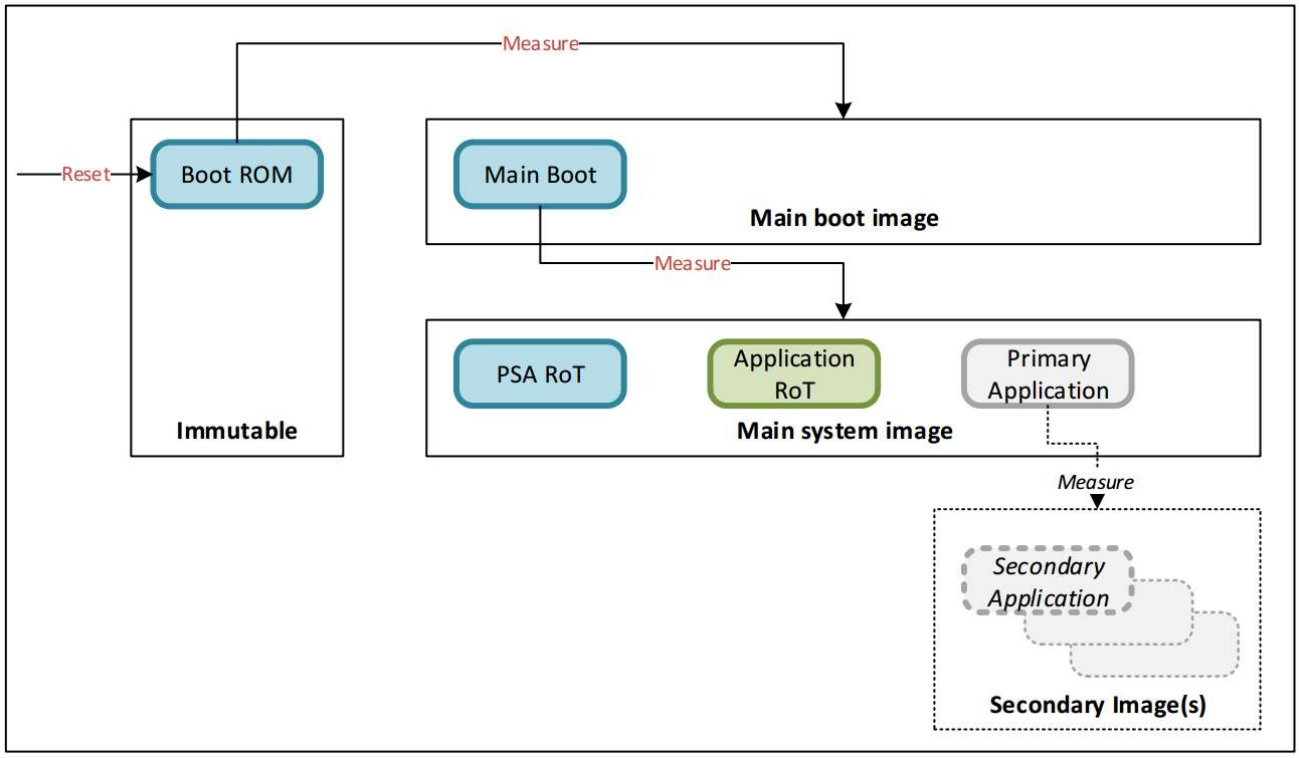
- 硬件级别的隔离环境

安全隔离环境多种实现方式，例如有的客户通过两个CPU的方式来实现隔离环境，一个CPU做为安全CPU，一个CPU作为非安全CPU，并且在总线上做出区分。另外一种方式是通过TrustZone，例如Cortex-M33本身就支持安全扩展，CPU有两个安全状态，可以理解为分时复用来支持隔离环境



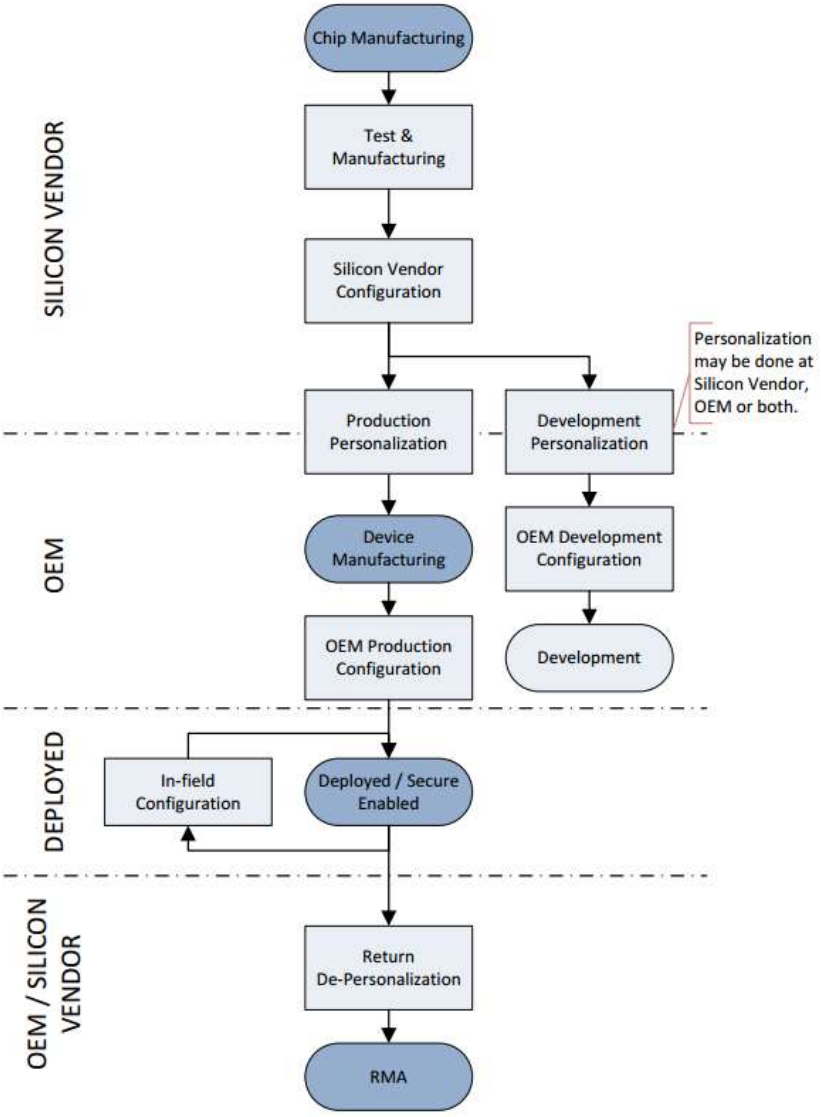

- 安全启动

安全启动需要从一个不可更改的信任根启动，保证代码的一致性，比较典型是从ROM启动




- 生命周期管理

芯片从生产到量产会经历不同的阶段，每个阶段都可能存在不同的敏感信息，例如秘钥或者代码等，需要有一个状态机来管理芯片的不同阶段，来设置不同的访问权限。




- 秘钥管理

秘钥是安全的核心，很多算法都是公开的，安全的保障是保护秘钥，在PSA的要求中一般是具备三个秘钥，分别是HUK，RoTPK，和attestation Key，HUK是每个芯片唯一的，可以理解为安全的种子来派生出其他秘钥；RoTPK是用来安全启动，可以用一个系列芯片用同一个秘钥，主要是实现安全启动的；还有个一个是attestation key是来实现设备和云端交互的认证。

| Name | On-Chip Data Size | Off-Chip Data Size | Access to On-Chip Data |
|---|---|---|---|
| ROTPK – RSA | 3072 bits (Key) | 0 bits | During boot ROM execution. Only |
| | 128 bits (Digest) | 3072 bits (Key) | During boot ROM execution only. |
| ROTPK – ECC | 256 bits (Key) | 0 bits | During boot ROM execution only. |
| HUK | 128 bits (key) | 0 bits | Trusted code/Trusted hardware only. |

从上面要求，我们可以看出PSA只是安全要求，与架构无关。无论是Arm的架构，还是非Arm的架构都可以满足PSA的安全要求，TrustZone只是实现PSA的一种快速方式。