

Software

Android tamper-resistant anti-replay secure storage solution and virtualization

Zhu, Bing

Open Source Technology Center (OTC)
Software and Services Group (SSG)

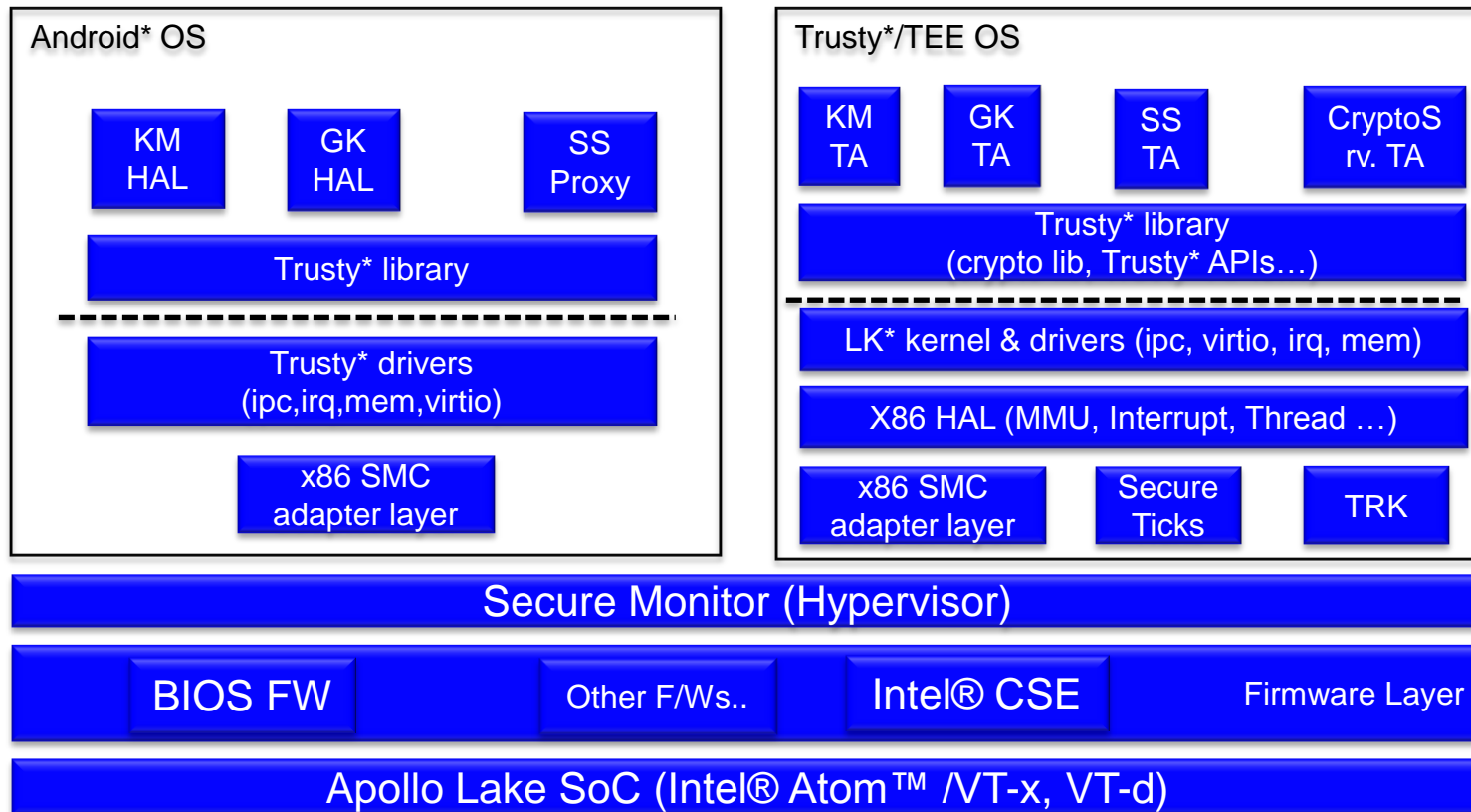
NOTICE & DISCLAIMER

- Intel technologies' features and benefits depend on system configuration

and may require enabled hardware, software or service activation.

- Performance varies depending on system configuration.
- Intel, the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.
- *Other names and brands may be claimed as the property of others.

Intel Android Automotive IVI Platform



Agenda

- Problem Statement
- Replay Protected Memory Block (RPMB)
- TEE/Trusty Secure Storage (SS)
- Secure Storage Virtualization on Hypervisor
- Future Directions

Problem Statement

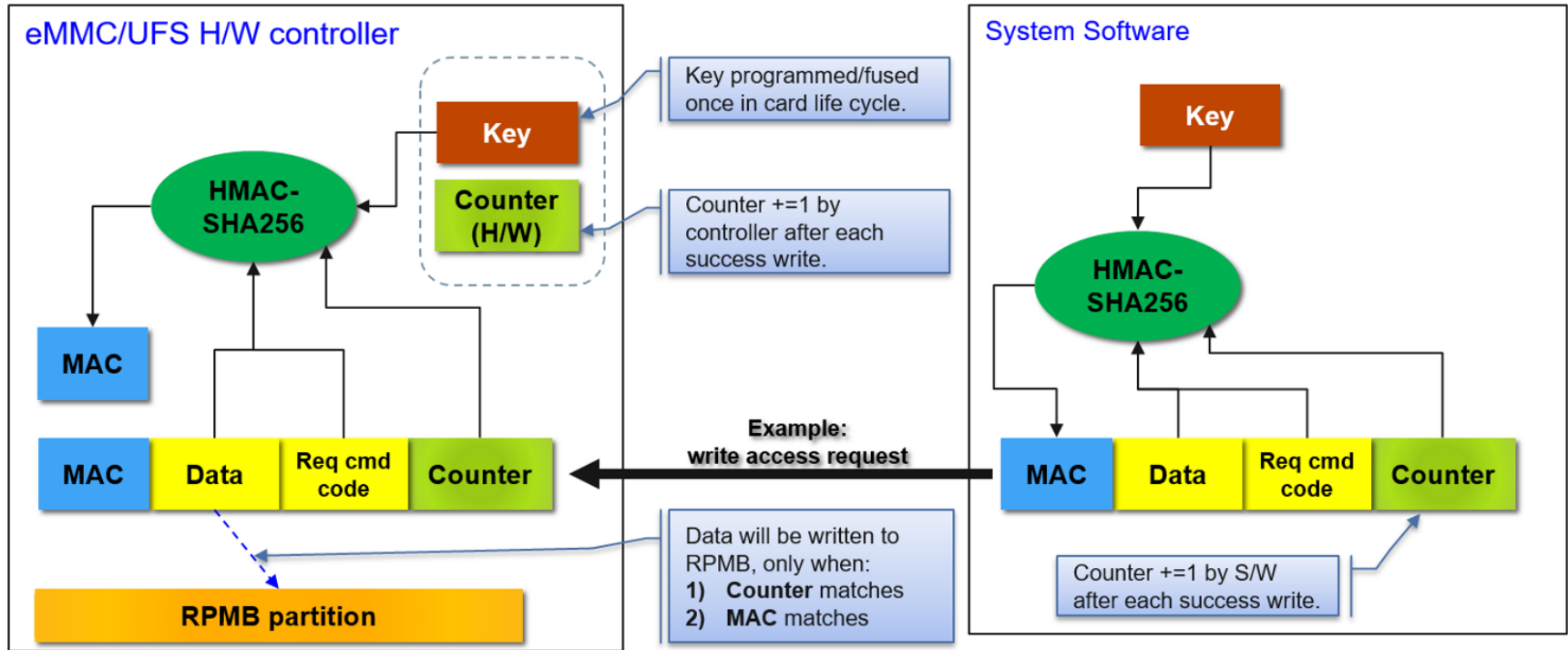
- Data security and privacy:
 - Screen-unlock (password/pin/pattern) attempt failure record for defending against brute force attack:
<https://source.android.com/security/authentication/gatekeeper>
 - The version of system image for preventing roll-back attack
 - Keybox (keypairs), e.g. for content protection and attestation
 - The templates of fingerprint or iris sensor images for authentication
- Google CDD requirements since Marshmallow :
 - [SR] STRONGLY RECOMMENDED/ SHOULD to use tamper-evident storage

Replay Protected Memory Block

Technical Details

- eMMC/UFS/NVMe have fixed physical RPMB partition(s) in device
 - pre-allocated during flash device manufacture.
- RPMB key can only be programmed once in its life time, and is invisible to any software as long as it is programed into h/w device.
- All data read/write request of access to RPMB will be authenticated by H/W RPMB controller with RPMB Authkey (Authentication Key):
 - Authenticate algorithm is HMAC-SHA256 (or 512)
 - H/W built-in monotonic Write Counter is used for replay-protection on WRITE access;
 - Software generated Random Number is used for replay-protection on READ access.
- Without RPMB Authkey, read access is still possible, but the data being read may not be authentic.

How it works (e.g. authenticated write access)



RPMB Key Generation and Programming

- RPMB Key generation requirements:
 - Key is tied to hardware unique key (HUK).
 - Key is also bound to eMMC/UFS/NVMe flash storage serial #.
- RPMB key programming:
 - Typically firmware is responsible for programming the RPMB Key (in cleartext) into RPMB controller through RPMB key programming interface.
 - Do it once in factory, or just right after eMMC/UFS replacement if applicable.
 - Key cannot be changed once it's programmed successfully (FUSED)

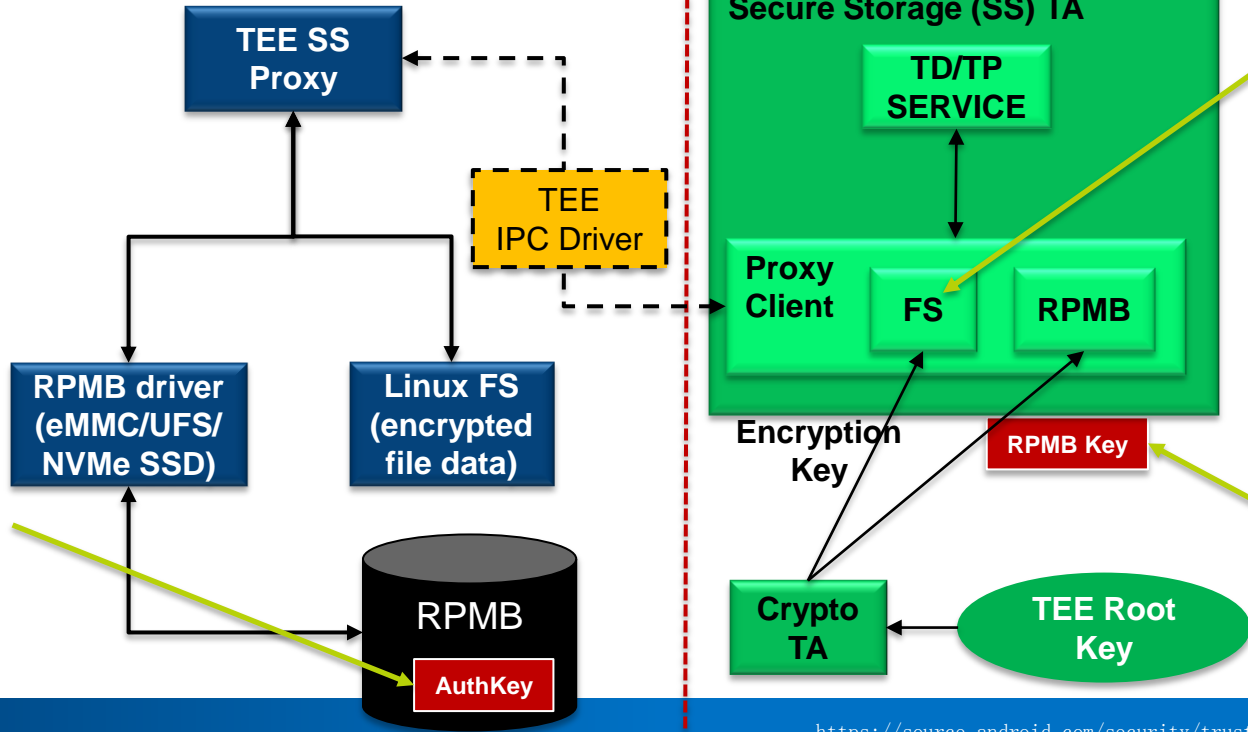
TEE/Trusty Secure Storage (SS) Architecture

Trusty*(TEE) Secure Storage

Linux/Android
(Non secure)

Trusty/TEE
(Secure)

Built-in
secure file
system

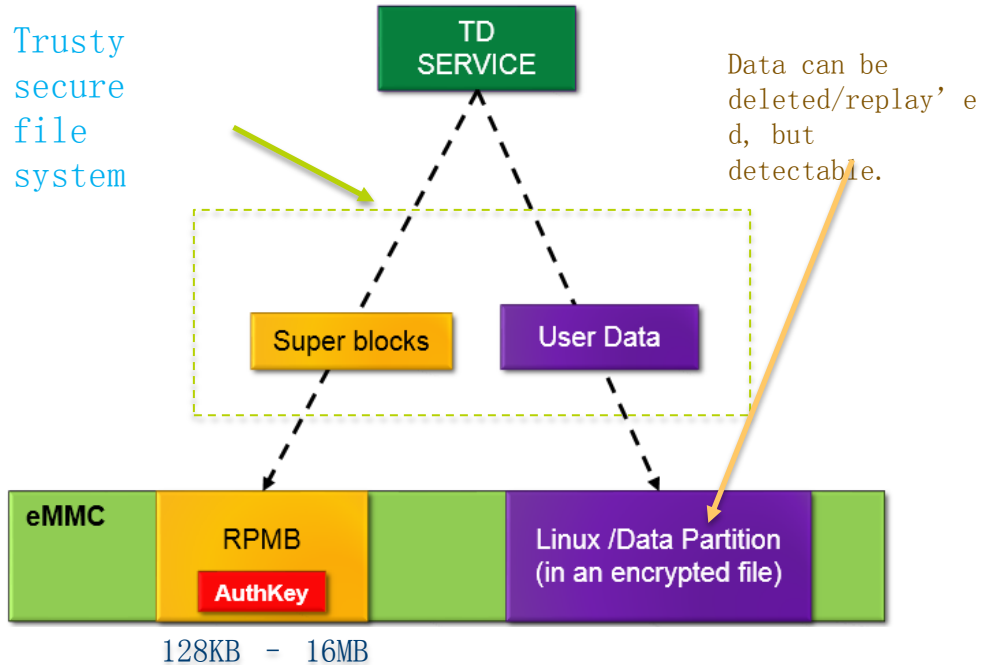


RPMB Key is factory-provisioned by firmware into flash device before production. Then it is invisible to any software.

RPMB Key is protected, never goes outside of TEE, and it is constantly generated/derived per each boot.

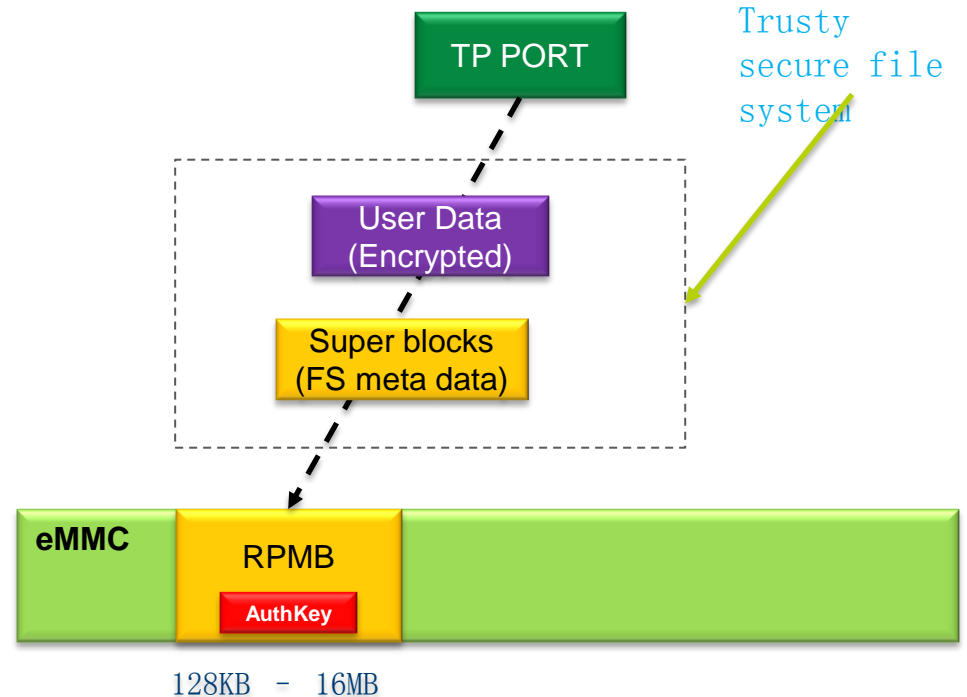
SS - Trusty TD Service: Tamper-Detection

1. File system meta-data is stored in RPMB
2. The user data encrypted with hardware-backed encryption key, is stored in Android/Linux-backed file system.
3. Support large amount of data.



SS - Trusty TP Service: Tamper-Proof

1. File system meta-data is stored in RPMB as well, and user Data also stored in RPMB
2. Size constrained; Typically 4MB, depending on eMMC/UFS/NVMe RPMB size.
3. Higher level of protection - Tamper Resistant!
4. Data survives in factory reset.



How to use (1/2)

Protection:
File created by one APP
can be accessible to any
other APPs (as long as
Selinux policy allow them
to access IPC driver)

Native Linux app to use
secure storage

SELinux policy protection for
access Trusty IPC driver
(/dev/trusty-ipc-dev0)

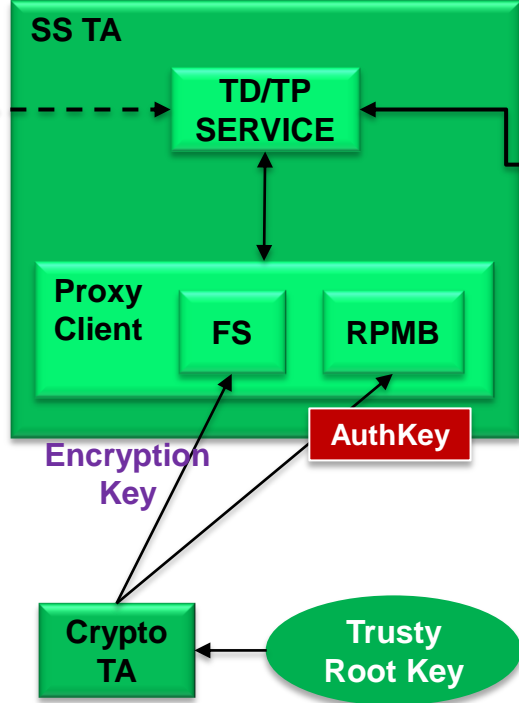
Android

OEM APP

SS Library
(C/C++)

Trusty
IPC Driver

Trusty

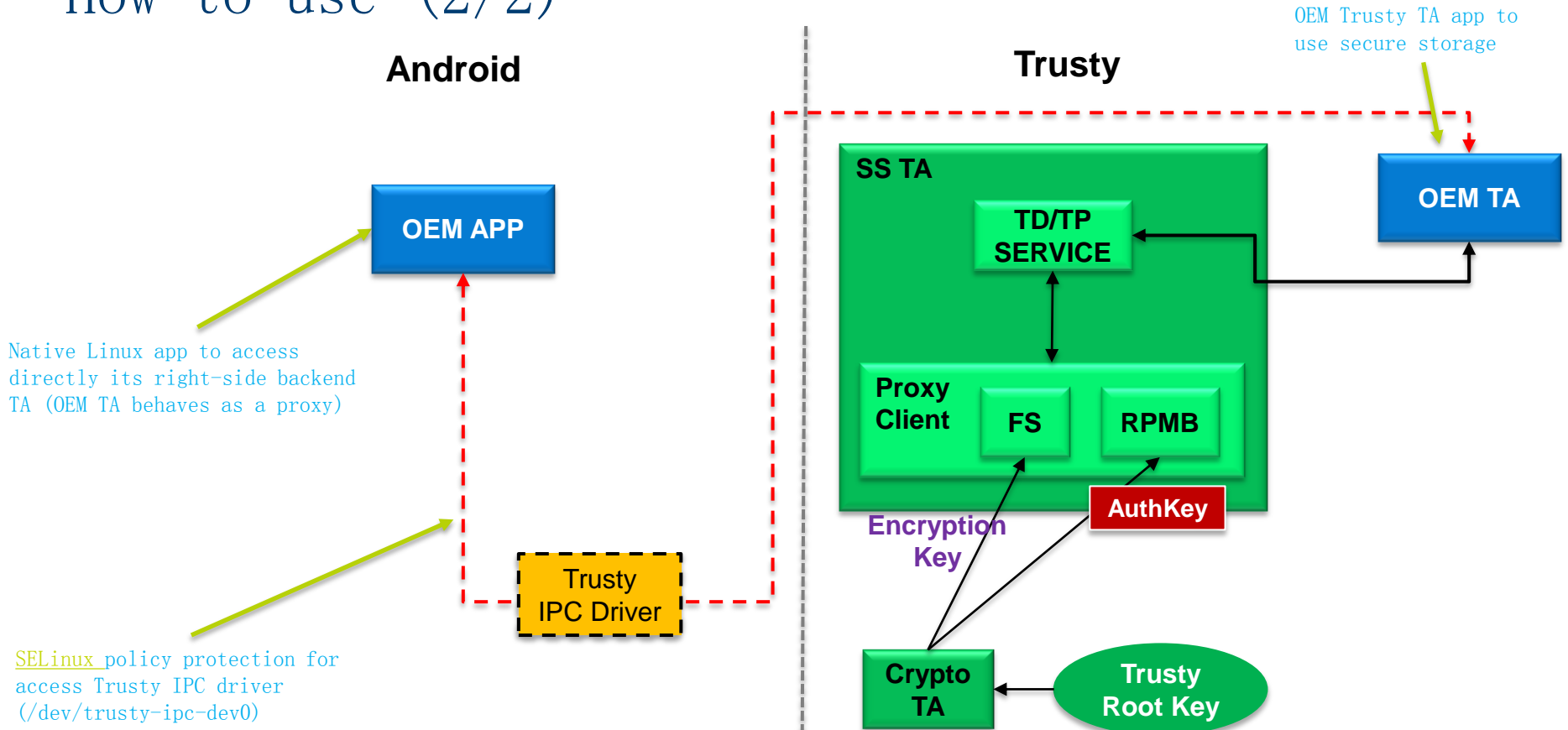


OEM Trusty TA app to
use secure storage

OEM TA

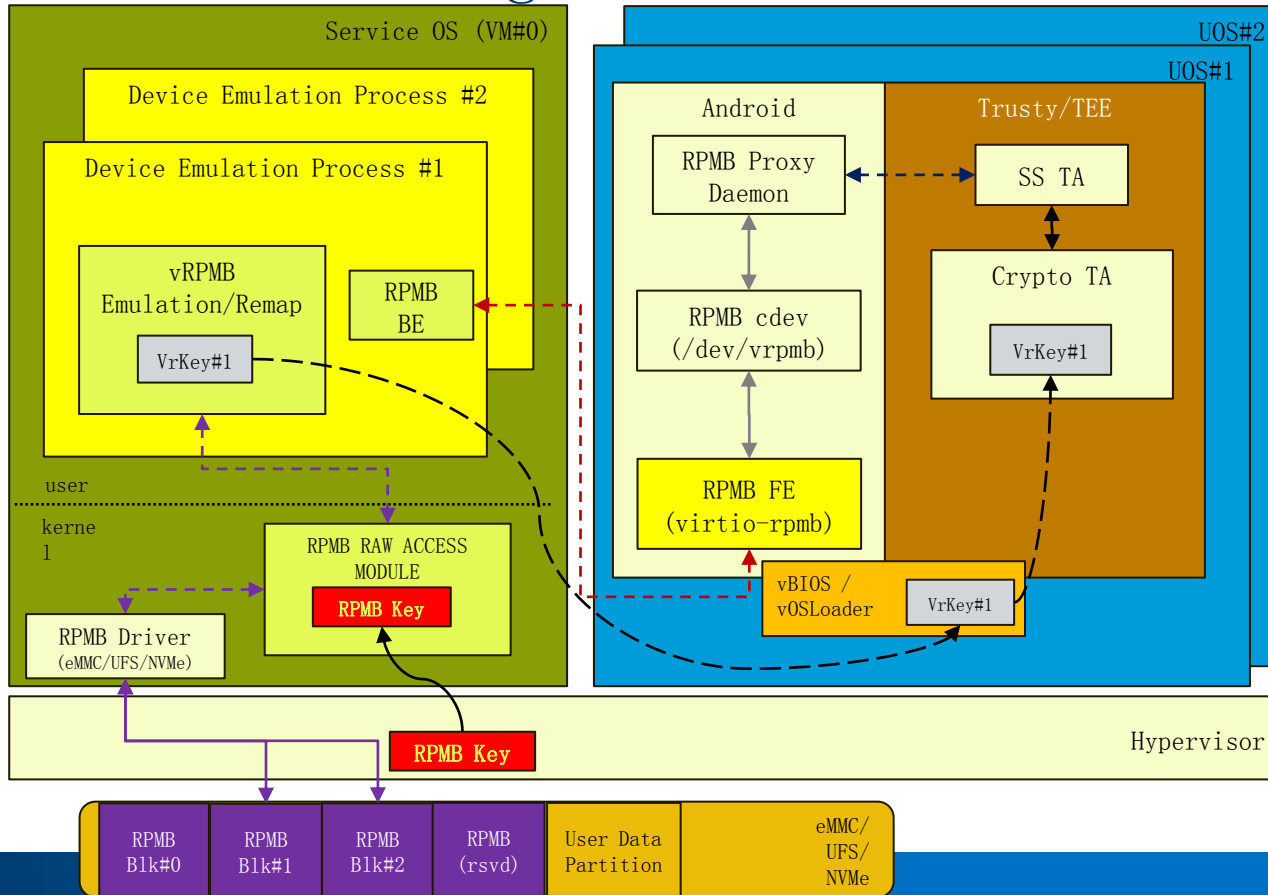
Protection:
File created by one
TA cannot be
accessible to any
other TAs.

How to use (2/2)



Secure Storage Virtualization on Hypervisor

Secure Storage Virtualization



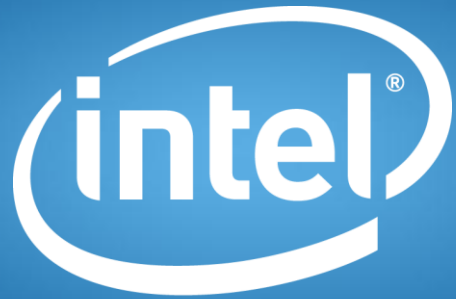
1. SOS (Service OS) is a closed system and privileged VM
2. The VrKey (virtual RPMB key) is generated randomly per UOS reboot, and securely distributed it to vSBL/vOSLoader/TEE.
3. Forward/remap vRPMB data/frame to physical RPMB partition.

Future Directions

Future Plan

- Multiple RPMB Targets / Partitions with H/W support
 - UFS3.0 supports 4 RPMB Partitions
 - NVMe storage supports MULTIPLE RPMB partitions

Q & A



Software