Linaro connect
San Francisco 2017

UEFI Secure Boot and DRI

Kalyan Kumar N

connect.linaro.org

Agenda

- Introduction
- RDK Boot Loader
- DRI (Disaster Recovery Image)
- RootFS Validation
- Build Environment

Introduction

- Standardization of the RDK set-top box firmware boot process
    - Increase industry awareness of UEFI/EDK2 solutions for set-top boot implementation

- Need secure boot with chain of trust with secure keys

- Implement RDK Bootloader and Disaster Recovery Image (DRI) requirements (use cases) using well defined standard.

RDK Boot Loader

- RDK BootLoader selects the valid Platform Code Image (PCI) from the Device non-volatile memory to load and execute.
- Enables secure boot by registering PK and KEK Key.
- LoadImage protocol of UEFI Boot service is used to load the kernel image from boot partition.
- kernel arguments are passed using Loadoptions of LoadedImageProtocol.
- Installs the FDT blob into EFI system configuration table

RDK Boot Loader

- FDT file will be read and install Using gFdtTableGuid.
- To maintain chain of trust key to validate Linux kernel can be placed in partition other than Boot partition.
- Key which validates Rootfs can be placed in Boot partition and will registred to UEFI variable and exported to Linux kernel.

RDK DRI

- RDK DRI reference implementation is UEFI application Resides in Flash memory and provides HTTP download method.

- Downloads PCI image file via Ethernet (USB to Ethernet interface) and store image into flash memory.

- DRI downloads Monolith Image comprised of 3 separate components
  1. Linux Kernel
  2. FDT file
  3. Root FileSystem

RDK DRI

- Linux and RootFs components are separately signed.
- Enabled USB host driver with support of Ethernet Adapter.
- Enabled Http drivers in Network Pkg.
- URL and IP details will be given to Http Driver
- After Downloading , Components are stored into different Flash partitions , and will validated prior to use.

RootFS validation

- After kernel secure boot , RDK wants rootfs also has to be authenticated.

- Kernel will bootup with temporary initramfs and will validates signed rootfs image and mount the same.

  Yocto build command for creating initramfs:

  INITRAMFS_IMAGE = "core-image-minimal-initramfs"

  INITRAMFS_FSTYPES = "cpio"

- Keys to validate Rootfs will be exported to kernel through UEFI .

RootFS validation

- Kernel provides EFIVAR file system, which enables accessing UEFI variables from kernel.

- Sign RootFS using openssl and generate sha256 hash file which will be part of monolithic image to verify the signature.

  Ex:
  ```
  openssl dgst -sha256 -sign "Key.key" -out  rootfs.sha256 rootfs.tar.bz2
  ```

- Once secure Linux kernel bootup with initramfs, it validate rootfs ,untar and mount the rootfs.

Build Environment

- EDK2 by default, does not support Hikey board.
- Openplatformpkg based on edk2 supports various development boards such as RPI, ARM juno etc. including HiKey
- OpenPlatformPkg can be added as an extra Pkg to edk2, to support UEFI for HiKey
- RdkPkg need to be added to EDK2, which provides application to access driver from OpenPlatformPkg and boot Linux kernel.

Build Environment

edk2:

https://github.com/tianocore/edk2.git

Hikey UEFI Firmware:

https://github.com/linaro-home/OpenPlatformPkg.git -b hikey-rdk

Rdk boot Loader and DRI

https://github.com/linaro-home/RdkPkg.git

RDK secure Boot Build setup script : Build script

Linaro
connect
San Francisco 2017

Thank You

#SFO17

SFO17 keynotes and videos on: connect.linaro.org

For further information: www.linaro.org