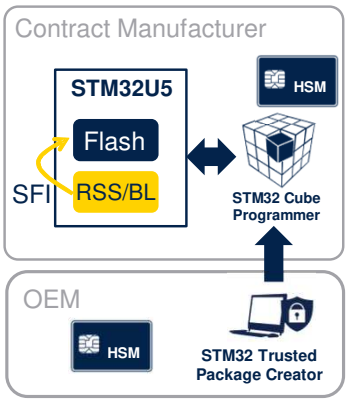
Hello, and welcome to this presentation describing the secure firmware install (SFI) feature offered by the Root Security Services.

Overview



- **RSS** is the secure part of the STM32 immutable bootloader (BL), Available when TrustZone is activated
- RSS provides services for (**SFI**) secure firmware install solutions

Application benefits

- Immutable root security services
- Enables Secure Firmware Install (SFI)

life.augmented

2

The Root Security Services (RSS) are the secure part of the STM32U5 immutable bootloader. They are available only when TrustZone is activated on the device.
RSS provides immutable root security services, used for example to run the STM32 secure firmware install (SFI) solution in an untrusted environment.
For more information on TrustZone and protected memories please refer to the on-line training module “STM32U5 Security Overview”.
A hardware security module (HSM) is in charge of:

- Securely storing OEM AES secret key
- Checking the STM32 device certificate that is used to authenticate STM32 device
- Generating and providing the license to the secure bootloader to securely install the encrypted firmware

on the STM32 device.

- Counting the number of produced STM32 devices.

RSS Key features

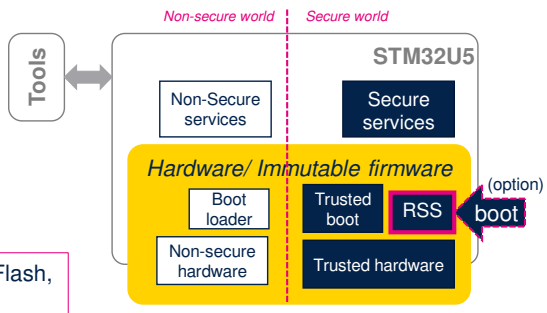
- **RSS** is a secure immutable firmware
  - Necessary to run STM32 **SFI**
  - Can be used as a unique entry point
  - RSS features a set of security services:
    - **Boot** or **run-time**

RSS boot time services

- Allocation of non-secure bootloader resources allocation (SRAM, Flash, peripherals, IOs, interrupts)
- Get/Set Flash secure option byte, used by the bootloader
- Get STM32U5 device certificate & certificate size

RSS run-time secure service

- Allows secure code running in the Flash HDP area to safely jump to a given address outside the protected area



life.augmented

When TrustZone is enabled, RSS is the secure immutable firmware supplied by ST during the production of the STM32U5 along with a unique key pair dedicated to the device.
After reset this immutable firmware is used as a unique entry point, featuring a set of security services that are available at boot time, and sometimes also at run-time.
RSS includes the necessary features to run the STM32 Secure Firmware Install (SFI) solution.
Boot time services of the RSS include:

- Non-secure bootloader resource allocation (SRAM, Flash, peripherals, IOs, interrupts)
- Functions to get or set Flash secure option bytes, used by the bootloader
- Function to get the STM32U5 device certificate and

its size, also used by the bootloader.

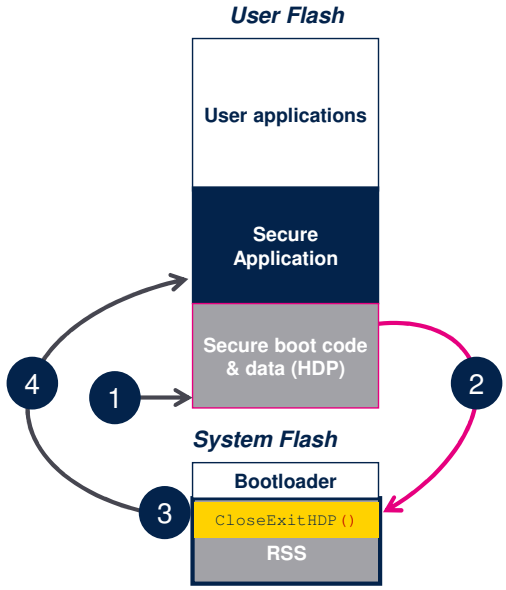
RSS also provides services for both secure and non-secure running firmwares.

One of those secure services allows secure firmware running in the Flash HDP area to safely jump to a given address outside the protected area.

All of these services are detailed in the next three slides.

RSS & Exiting HDP area

- The HDP area is activated by secure code, setting the HDPEN option bit
- The typical sequence to jump outside of it:
  1. Device boots, executing sensitive code in the HDP area
  2. Boot code calls the HDP exit function in RSS lib
  3. RSS hides the secure HDP area until the next reset, then branches to the (secure) application code
  4. Secure firmware can no longer access the HDP area
- Refer to **RM0456** for more details on the HDP close & exit function



life.augmented

4

Secure Hide protection (HDP) is an additional protection mechanism within the TrustZone secure domain.
The code embedded in HDP is executed first, and at the end of its execution, it jumps to the secure user application.
The code and data protected by HDP is longer accessible until the next system reset.
The HDP area is activated by secure code, setting the HDPEN option bit.
Let's now explain how RSS helps jumping outside the HDP area.
The boot code embedded in HDP executes after reset. See step one in the figure.
At the end of its execution, it calls RSSLIB_sec_CloseExitHDP. This is step two.

Then this RSSLIB_sec_CloseExitHDP function closes the Flash HDP secure memory area, this is the step 3, and jumps to the reset handler pointed to by the vector table whose address is passed as input parameter.
This is the step four.
This function resets the STM32U5 in case of failure, due to bad input parameter value for example.

Secure firmware install (SFI)

- Secure firmware install (SFI):
  - Allows secure and counted installation of OEM firmware in untrusted production environments (such as OEM contract manufacturer)
  - SFI is implemented using the secure RSS and the non-secure immutable bootloader
  - OEM firmware protected by SFI can be stored in embedded flash or encrypted in external flash
  - The number of STM32 devices on which the firmware has been installed can be counted by the HSM
- When external Flash memory is targeted by SFI, OEM firmware is encrypted with an external firmware and the data AES key
  - The OTFDEC Peripheral can be used to accelerate encryption
  - SFI can re-encrypt OEM external firmware using the AES key(s) dedicated to the OTFDEC peripheral
  - Keys can be globally managed (by the tools), or they can be device specific (e.g. locally computed using the true RNG peripheral)

life.augmented

5

Secure firmware install (SFI) is a global solution for STM32U5 Series of microcontrollers, allowing secure and counted installation of OEM firmware in untrusted production environments, such as OEM contract manufacturer.
SFI is implemented using the secure RSS and the non-secure immutable bootloader.
OEM firmware protected by SFI can be stored in the device's embedded flash or encrypted in external flash connected via OCTOSPI.
Authenticity, integrity and confidentiality of the OEM internal firmware (and option bytes) are checked before embedded Flash is programmed with decrypted firmware (and option bytes).
The STM32U5 SFI solution consists in having the entire

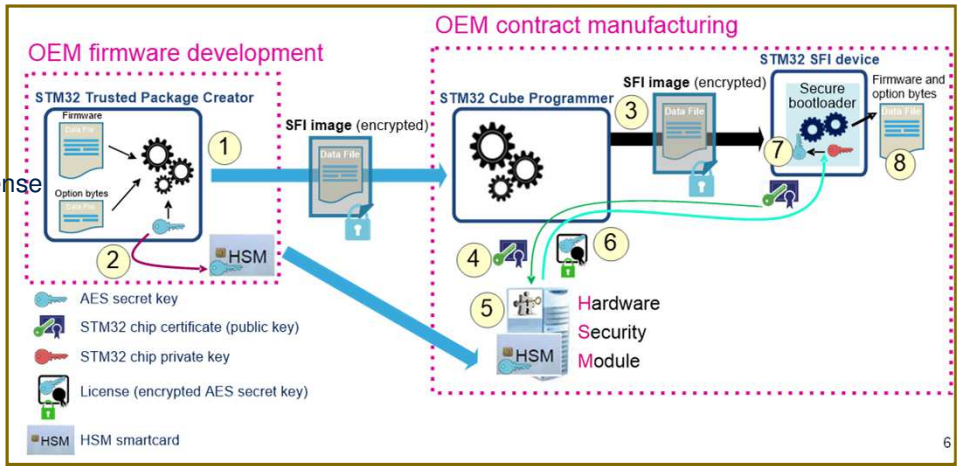
OEM firmware and option bytes encrypted with an AES secret key, thanks to STM32 Trusted Package Creator tool.
This is done during the development of the OEM firmware.
Confidentiality of this AES secret key is ensured using a unique key pair dedicated to the STM32 device, with the private key readable only by RSS.
Refer to AN5391, entitled STM32L5/U5 SFI tools, bootloader and RSS interface for more details.
When external Flash memory connected via OCTOSPI is targeted by SFI, the OEM firmware code must be encrypted with an external firmware and the data AES key.
This key can be:

- Common to all devices (in this case tools could perform the encryption), or
- Unique per device (in this case firmware is encrypted inside the device).

Note that on-the-fly decryption of encrypted firmware stored in SPI flash memories is only available on STM32U5 devices.
For more details, please refer to the end of this module or the application note AN4992 for secure firmware install (SFI) solutions.

SFI to internal flash

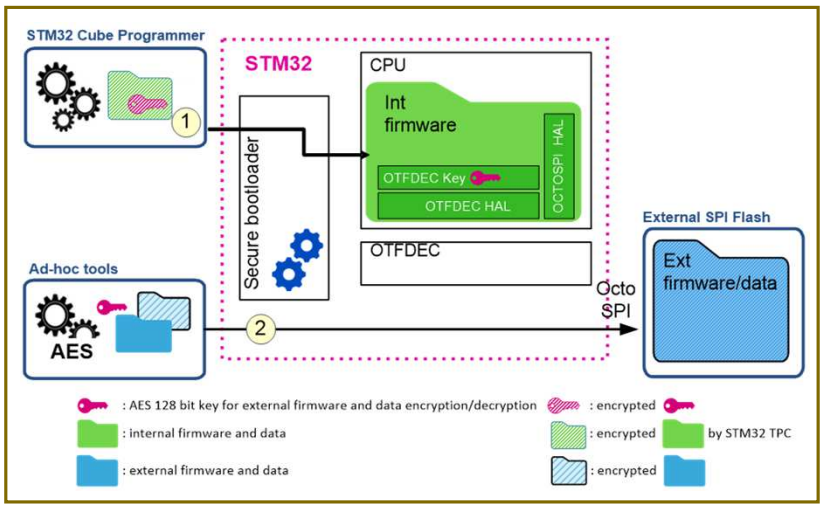
1. SFI image (encrypted) available from *STM32 Trusted Package Creator*
2. The OEM programs the HSM with the AES secret key
3. Start of the SFI process
4. Device certificate retrieval
5. STM32 device authentication in the HSM
6. The HSM provides the license to the STM32
7. The RSS retrieves the OEM AES secret key encrypted in the license
8. The encrypted firmware and option bytes are decrypted then programmed



The installation of secure firmware in internal Flash memory goes as follows (the numerical steps are shown in the diagram):
(1) The (encrypted) SFI image is available from STM32 Trusted Package Creator
(2) The OEM programs the HSM with the AES secret key
(3) Start of the SFI process
(4) Device certificate retrieval
(5) STM32 device authentication in the HSM
(6) The HSM provides the license to the STM32 device
(7) The RSS retrieves the OEM AES secret key encrypted in the license
(8) Encrypted firmware and option bytes are transferred, decrypted then programmed.

SFI to internal & external flash (1)

1. Secure programming of internal Flash memory, using SFI
2. Encryption then programming of external firmware and data
   - ✓ See next slide for more details



NB: The OTFDEC peripheral is only available in STM32U5 devices

life.augmented

7

The cryptographic engine responsible for the on-the-fly external Flash memory decryption (OTFDEC) supports the AES standard cryptographic algorithm.
Thanks to this standard algorithm, the OEM can encrypt the external firmware and data on the host before programming it into the external Flash memory, without using the STM32 secure bootloader.
This slide shows that the secure programming of internal Flash memory (1) and the encryption plus programming of the external firmware and data (2) can be done in two separate flows.
The first flow uses the secure bootloader, while the second uses the OEM host to program the external Flash memory.
Then, during each secure boot, the secure internal

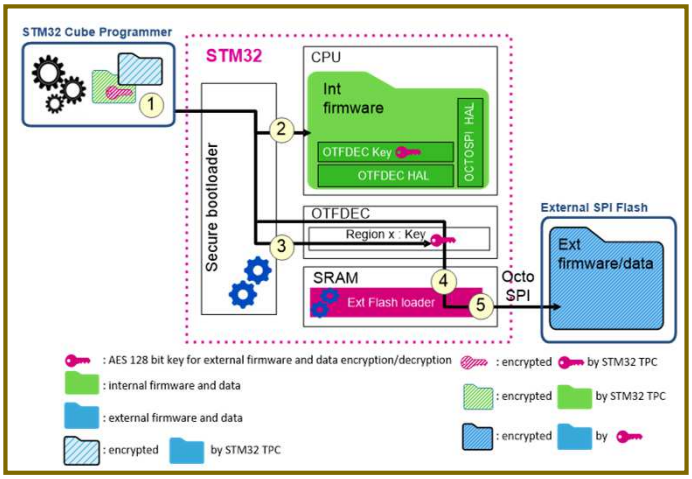
firmware first copies the AES firmware and data key(s) into write-only OTFDEC key registers, then activates the OTFDEC region tied to those keys.
At this point the CPU can seamlessly read data and fetch code from external Flash memory once the OCTOSPI driver has been initialized.

SFI to internal & external flash (2)

1. Create an SFI image, with:
   - Internal firmware and data (including external Flash memory drivers)
   - External firmware and the data *AES key*
   - External firmware and data
2. Internal Flash memory programming
3. External firmware and data *AES key* programming in the OTFDEC peripheral
   - Alternatively, such key(s) can be managed locally in the device, not globally in the flashing tools.
4. External Flash memory chunk encryption
   - Not required if the image is already encrypted with the AES key

NB: The OTFDEC peripheral is only available in STM32U5 devices

5. External Flash memory programming by the user's firmware



TPC: Trusted Package Creator

This slide represents the sequence where the STM32 secure bootloader handles both internal firmware installation and external firmware installation with a global external Flash memory AES key and the help of an external Flash memory loader. The numerical steps are shown in the diagram.
(1) Create an SFI image, with a) internal firmware and data (including external Flash memory drivers), b) external firmware and the data AES key, and c) external firmware and data
(2) Internal Flash memory programming, as described on the previous slide.
(3) External firmware and data AES key programming in the OTFDEC peripheral. Alternatively to what is drawn on the slide, this key can be managed locally in the

device, not globally in the flashing tools.
(4) External Flash memory chunk encryption, required if the image was not encrypted by the STM32 Trusted Package Creator
(5) External Flash memory programming by the user's firmware
Then, during each secure boot, the secure internal firmware first copies the AES firmware and data key(s) into write-only OTFDEC key registers, then activates the OTFDEC region tied to those keys.
At this point the CPU can seamlessly read data and fetch code from external Flash memory once the OCTOSPI driver has been initialized.

References

- For more details and additional information, refer to the following
  - RM0456: STM32585x and STM32U575x Reference Manual.
  - AN2606: “STM32 microcontroller system memory boot mode”
  - AN4992: Overview of secure firmware install (SFI)
  - UM2237: STM32CubeProgrammer software description
  - UM2238: STM32 Trusted Package Creator software description
  - AN5391: STM32L5/U5 SFI tools, bootloader and RSS interface

life.augmented

For more details, please refer to:

- Application note AN2606 about the STM32 microcontroller system memory boot mode
- Application note AN4992 about the Overview of secure firmware install (SFI)
- User manuals for the STM32CubeProgrammer and STM32 Trusted Package Creator are also available on the ST website.

Thank you

life.augmented

Thank you for attending this presentation!
You can also refer to the following presentations on STM32U5 security features:

- Security overview
- Enhanced anti-tamper
- Enhanced key storage
- Hash and random number
- Symmetric crypto
- Asymmetric crypto
- Crypto lib
- Security certification.