AN4992
Application note

Overview secure firmware install (SFI)

Introduction

Outsourcing of product manufacturing enables original equipment manufacturers (OEMs) to reduce their direct costs and concentrate on high added-value activities such as research and development, sales and marketing.

However, contract manufacturing puts the OEM's proprietary assets at risk, and since the contract manufacturer (CM) manipulates the OEM's intellectual property (IP), it might be disclosed to other customers, or appropriated.

To meet the new market security requests and protect customers against any leakage of their IPs, STMicroelectronics introduces a new security concept, the Secure Firmware Install (SFI), allowing to program OEM firmware into STM32 internal Flash memory in a secure way (i.e with confidentiality, authentication and integrity checks).

SFI solution is introduced on the microcontrollers listed in *Table 1*.

STM32 Series supports protection mechanisms allowing to protect critical operations (such as cryptography algorithms) and critical data (such as secret keys) against unexpected access.

This application note gives an overview of the STM32 SFI solution with its associated tools ecosystem and explains how to use it to protect OEM firmware during the CM product manufacturing stage.

Table 1. Applicable products

Type	Part numbers	Order code
Microcontrollers	STM32H753xI	All order codes supported (refer to datasheet ordering information section)
	STM32L462CE	STM32L462CEU6F(1)

1. This is the only supported order code. This code is not listed in the datasheet ordering section. Contact ST sales representative (special order).

Contents

List of figures

1 Preamble

1.1 Related documents

Refer to the following documents available from *www.st.com* (unless an NDA applies):

[AN3155] USART protocol used in the STM32 bootloader

[AN3156] USB DFU protocol used in the STM32 bootloader

[AN4286] SPI protocol used in the STM32 bootloader

[AN5054] Secure programming using STM32CubeProgrammer(a)

[AN5243] STM32H7 bootloader SFI security extension(a)

[AN5251] STM32L4 bootloader SFI security extension(a)

[RM0394] STM32L41xxx/42xxx/43xxx/44xxx/45xxx/46xxx advanced Arm®-based 32-bit MCUs(b)

[RM0433] STM32H743/753 and STM32H750 advanced ARM®-based 32-bit MCUs(b)

[UM2237] STM32CubeProgrammer software description

[UM2238] STM32 Trusted Package Creator tool software description

[UM2508] Hardware secure module (HSM) for STM32CubeProgrammer secure firmware install (SFI) specification(a)

arm

a. Available under NDA.

b. Arm is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

1.2 Glossary

Table 2. Glossary terms

Term	Definition
AES	Advanced encryption standard
AES GCM	AES Galois counter mode
CM	Contract Manufacturer
FT	Flash programming tool
HSM	Hardware security module
MAC	Message authentication code
MCU	Microcontroller unit
OB	Option bytes
OEM	Original equipment manufacturer
RDP	Readout protection
Secure boot	Root of trust, check STM32 security protection
Secure bootloader	Standard ST bootloader with additional security features
SFI	Secure firmware install
WRP	Write protection

2 STM32 secure firmware install (SFI)

2.1 SFI principles overview

SFI is a secure mechanism implemented in STM32 microcontrollers that allows secure and counted installation of OEM firmware in untrusted production environment (such as OEM contract manufacturer). SFI is implemented in a secure bootloader.

The SFI process prevents the OEM firmware code from:

- being accessed by the contract manufacturer.
- being extracted or disclosed.

This mechanism consists in having the whole OEM firmware and the option bytes encrypted with an AES secret key, thanks to STM32 Trusted Package Creator tool(1), during OEM firmware development.

OEM must use STM32 Trusted Package Creator tool to program HSM with its own AES secret key(2), its own nonce, and a maximum installation counter.

OEM contract manufacturer have to use STM32CubeProgrammer to initiate SFI process and send encrypted SFI image(3) to STM32 device.

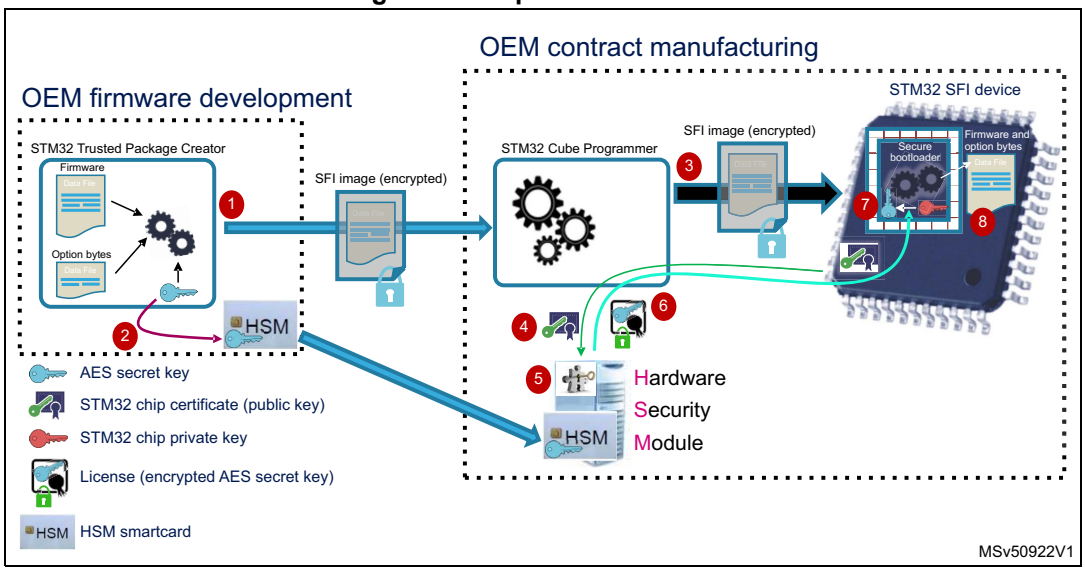
A hardware security module (HSM) is in charge of:

- Securely storing OEM AES secret key
- Checking STM32 device certificate(4) that is used to authenticate STM32 device(5)
- Generating and providing the license(6) to the secure bootloader to securely install the encrypted firmware on STM32 device.
- Counting number of produced STM32 devices.

The applicable STM32 microcontrollers are provisioned by STMicroelectronics with device dedicated public/private keys (unique key pair per device). The device keys can be accessed only through the embedded secure bootloader that retrieve AES secret key(7) by decrypting license using device private key.

Thanks to STM32 security features and cryptographic algorithm, STM32H753xI and STM32L462CE microcontrollers support secure OEM firmware programming and ensure OEM firmware protection (confidentiality, authenticity and integrity) during OEM-CM manufacturing stage. The secure firmware install solution securely receives and decrypts the firmware and option bytes inside STM32 internal Flash memory (8).

Figure 1. SFI process overview



1. SFI image (encrypted) available from STM32 Trusted Package Creator.
2. Program HSM with AES secret key.
3. SFI process launch.
4. Device certificate.
5. STM32 device authentication.
6. Provide license.
7. Retrieve AES secret key.
8. Firmware and option bytes programming.

The secure bootloader is a standard ST bootloader with additional security features.

If the STM32 microcontroller is reset during retrieving AES secret key(7), all sensitive data are erased before restarting initial SFI procedure.

During SFI process, the secure bootloader never allows any other code to access user Flash memory or SRAM.

2.2 SFI security features

The SFI security features are the following:

- Only genuine STMicroelectronics STM32 microcontrollers can install the protected firmware.
- The number of STM32 devices on which the firmware has been installed can be counted.
- Authenticity, integrity and confidentiality of the OEM firmware and option bytes are checked and user Flash memory is programmed with decrypted firmware and option bytes.

3 STM32 secure bootloader

The STM32 secure bootloader, implementing SFI, is programmed by STMicroelectronics during STM32 manufacturing. Its main task is to manage protocol communication between STM32CubeProgrammer and STM32 device in order to:

- identify STM32 device
- exchange device certificate and license
- download SFI encrypted image inside STM32

3.1 STM32H753xI

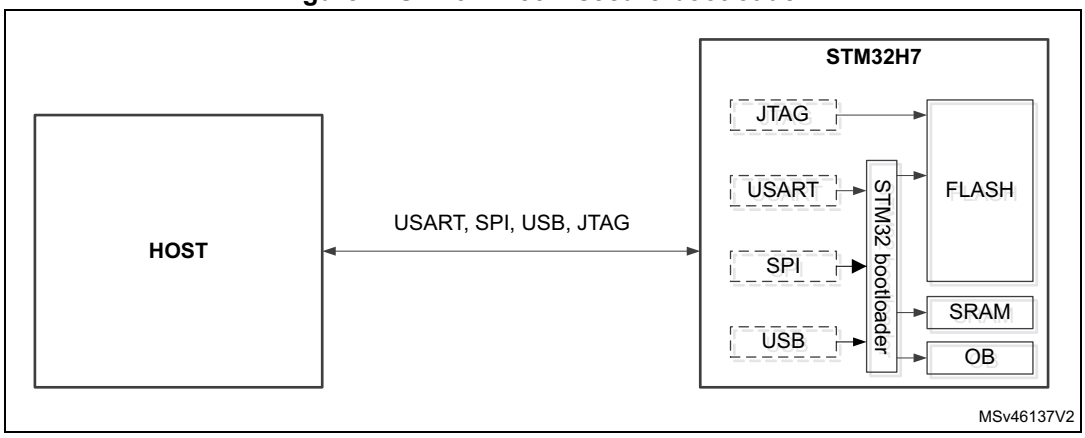
3.1.1 Secure bootloader overview

On the STM32H753xI, the secure bootloader is stored in the internal boot ROM (system memory) and supports following interfaces: USART, SPI, USB-DFU and JTAG.

The STM32H753xI secure bootloader allows to run the SFI process several times after complete erase of the internal user Flash memory, if erase allowed by installed application.

For more details on STM32 bootloader protocols, refer to *[AN3155]* (USART protocol), *[AN4286]* (SPI protocol) and *[AN3156]* (USB DFU protocol). For STM32H753xI bootloader SFI security extension, refer to *[AN5243]* (under NDA). The documents are available on *www.st.com* (unless an NDA applies).

Figure 2. STM32H753xI secure bootloader



3.1.2 User Flash memory mapping

On STM32H753xI, since the secure bootloader is stored in the system memory, the product is delivered in RDP level0 and all the user Flash memory is available for OEM.

The OEM firmware can be built to start at the beginning of the user Flash memory (starting address 0x0800 0000).

3.1.3 Secure boot path

After successful SFI process and if a secure area is set, the STM32H753xI always boot in the secure area closest to the configured boot address.

More information is available in *[RM0433]*.

3.2 STM32L462CE

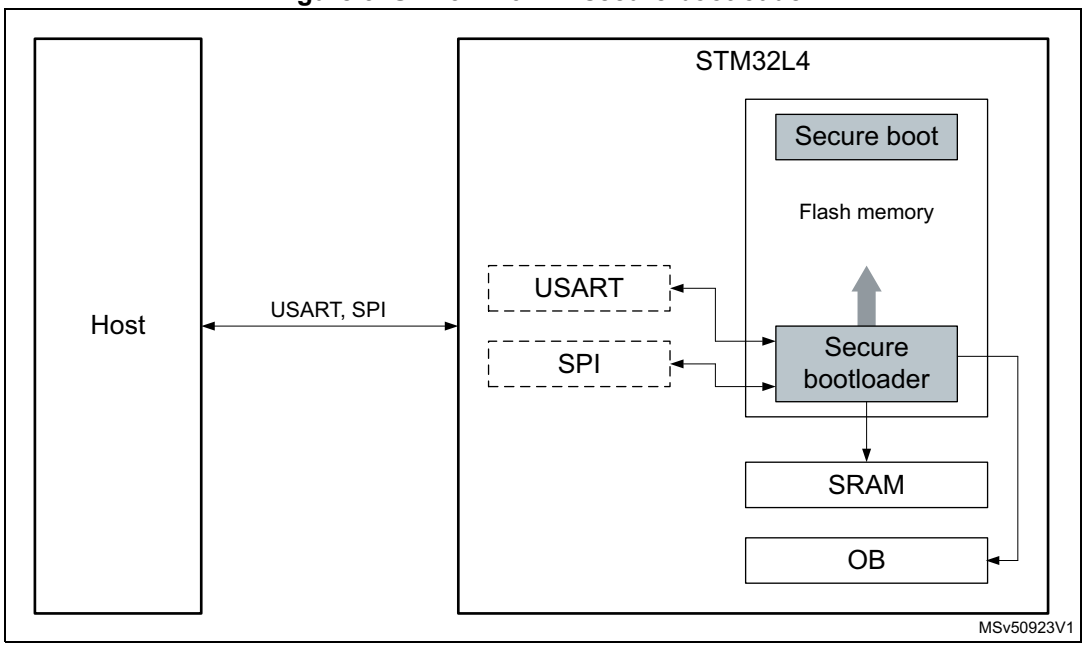
3.2.1 Secure bootloader overview

On the STM32L462CE[a], the secure bootloader is stored in internal user Flash memory and supports USART and SPI serial interfaces.

After successful SFI process (no error detected during firmware and OB installation), the secure bootloader is erased from user Flash memory and the SFI cannot be launched again afterwards. In case of any error during SFI process (authentication, integrity, power down,...), it is possible to run again SFI after an automatic cleaning process.

For more details on STM32 bootloader protocols, refer to *[AN3155]* (USART protocol) and *[AN4286]* (SPI protocol). For STM32L462CE bootloader SFI security extension, refer to *[AN5251]* (under NDA). These documents are available on *www.st.com* (unless an NDA applies).

Figure 3. STM32L462CE secure bootloader



a. The only supported order code is STM32L462CEU6F. This code is not listed in the datasheet ordering section. Contact ST sales representative (special order).

3.2.2 User Flash memory mapping

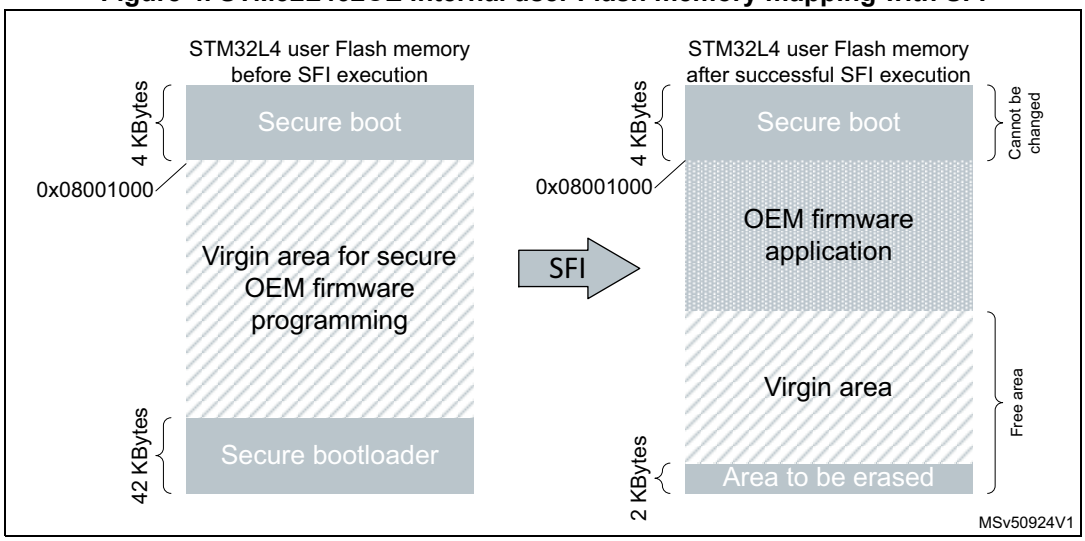
The secure bootloader on STM32L462CE[a] uses the following part of the internal user Flash memory which is protected in RDP level1:

- 4 Kbytes at the beginning of the user Flash (secure boot)
- 42 Kbytes at the end of the user Flash (secure bootloader)

The STM32L462CE secure boot area is a non-modifiable code area which is executed on Reset. It checks the security configuration (RDP and WRP on secure boot) and that no error occurs during SFI installation.

After successful SFI execution, the last 42 Kbytes of the Flash memory are available for OEM application (40 Kbytes erased by SFI process and last 2 Kbytes can be erased by OEM firmware). Only the first 4 Kbytes (secure boot) are kept in internal user Flash and cannot be changed, as a consequence the OEM firmware must be built with an offset of 4 Kbytes (starting address 0x08001000).

Figure 4. STM32L462CE internal user Flash memory mapping with SFI



3.2.3 Secure boot path

In order to always boot in user Flash memory, it is highly recommended to keep the default option bytes configuration with nSWBOOT0 set to 0 and nBOOT0 set to 1.

More information is available in *[RM0394]*.

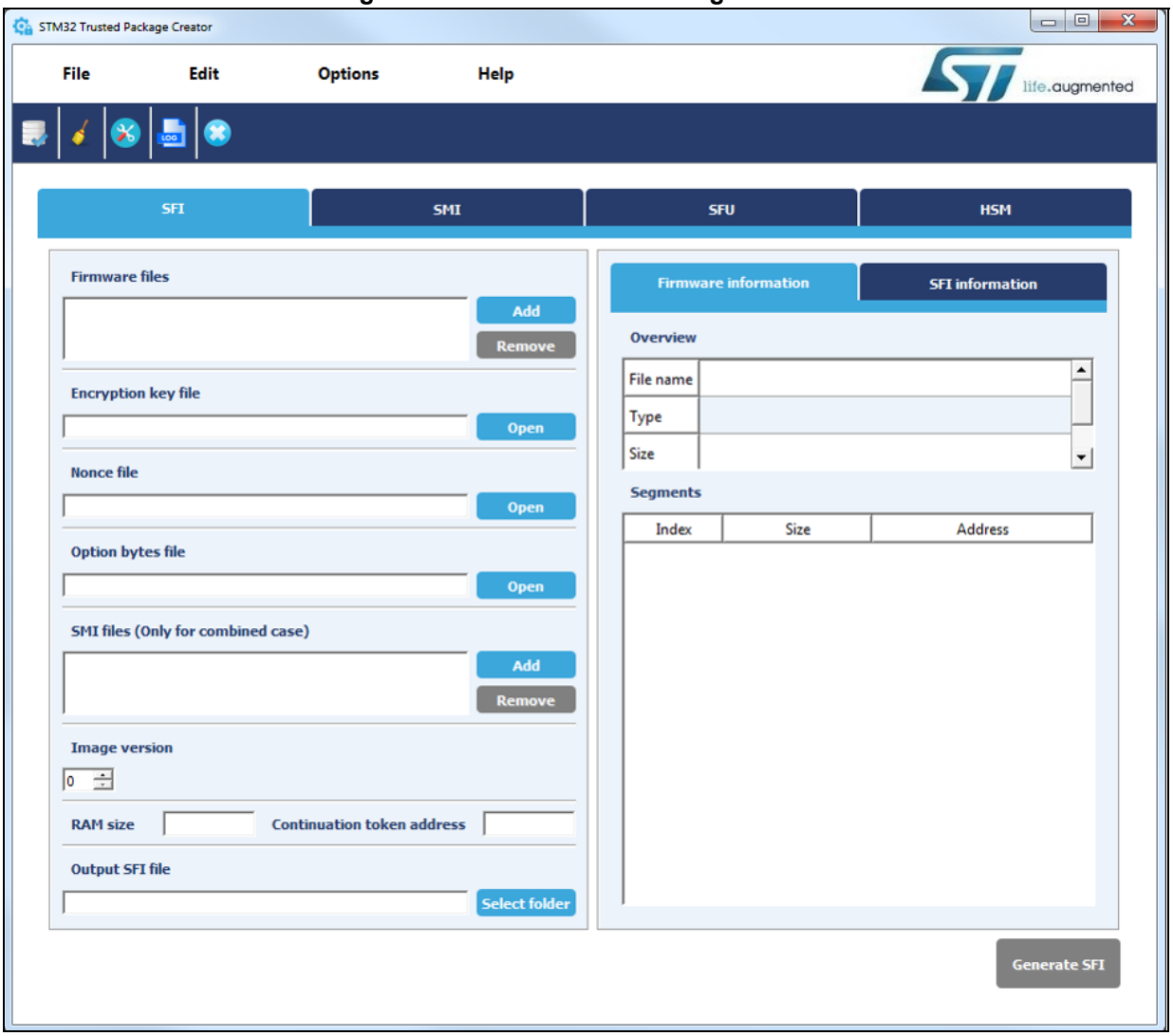
a. The only supported order code is STM32L462CEU6F. This code is not listed in the datasheet ordering section. Contact ST sales representative (special order).

4 SFI image preparation

The SFI image can be prepared by using STMicroelectronics STM32 Trusted Package Creator available both in CLI (command line interface) and GUI (graphical user interface) modes. This tool can be downloaded free of charge from *www.st.com*.

It allows the generation of SFI images for STM32 microcontrollers.

Figure 5. STM32 Trusted Package Creator



4.1 SFI firmware image format

The SFI format is an encryption format for firmware created by STMicroelectronics. It uses AES-GCM algorithm with a 128-bit key to transform a firmware in Elf, Hex, Bin or Srec formats into an encrypted and authenticated firmware in SFI format.

An SFI firmware image is composed of a header plus several areas. The areas are usually contiguous firmware areas. The last area is the configuration area containing the option byte values to be programmed when the SFI is complete.

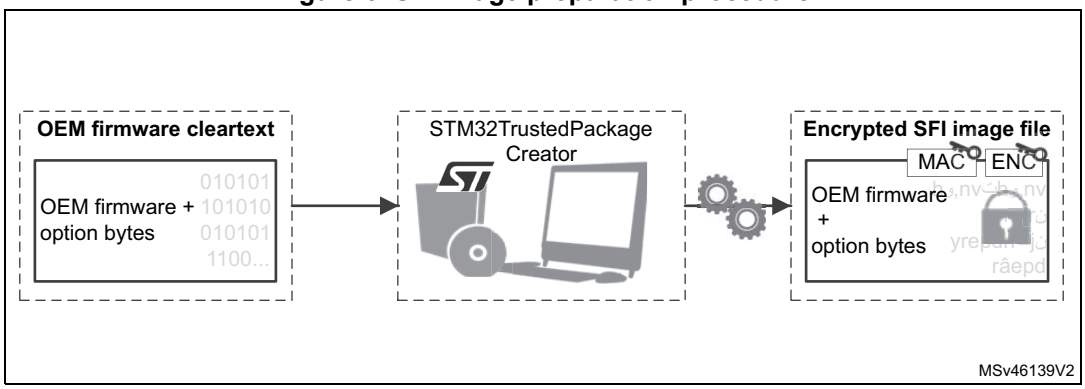
4.2 SFI firmware image creation procedure

To obtain SFI firmware images in the correct format (see *Section 4.1: SFI firmware image format*), the OEM must follow the steps below to build an SFI image:

1. Build the firmware image(s) using regular development tools.
2. Generate the binary image(s). A binary image does not need to be contiguous (i.e. it can cover multiple disjoint areas).
3. On STM32L462CE, binary image(s) must start with an offset of 4 Kbytes (starting address 0x08001000) from the beginning of the internal Flash and must not use the last 42 Kbytes of the internal user Flash memory.

Figure 6 shows the SFI image preparation procedure based on the STM32 Trusted Package Creator.

Figure 6. SFI image preparation procedure

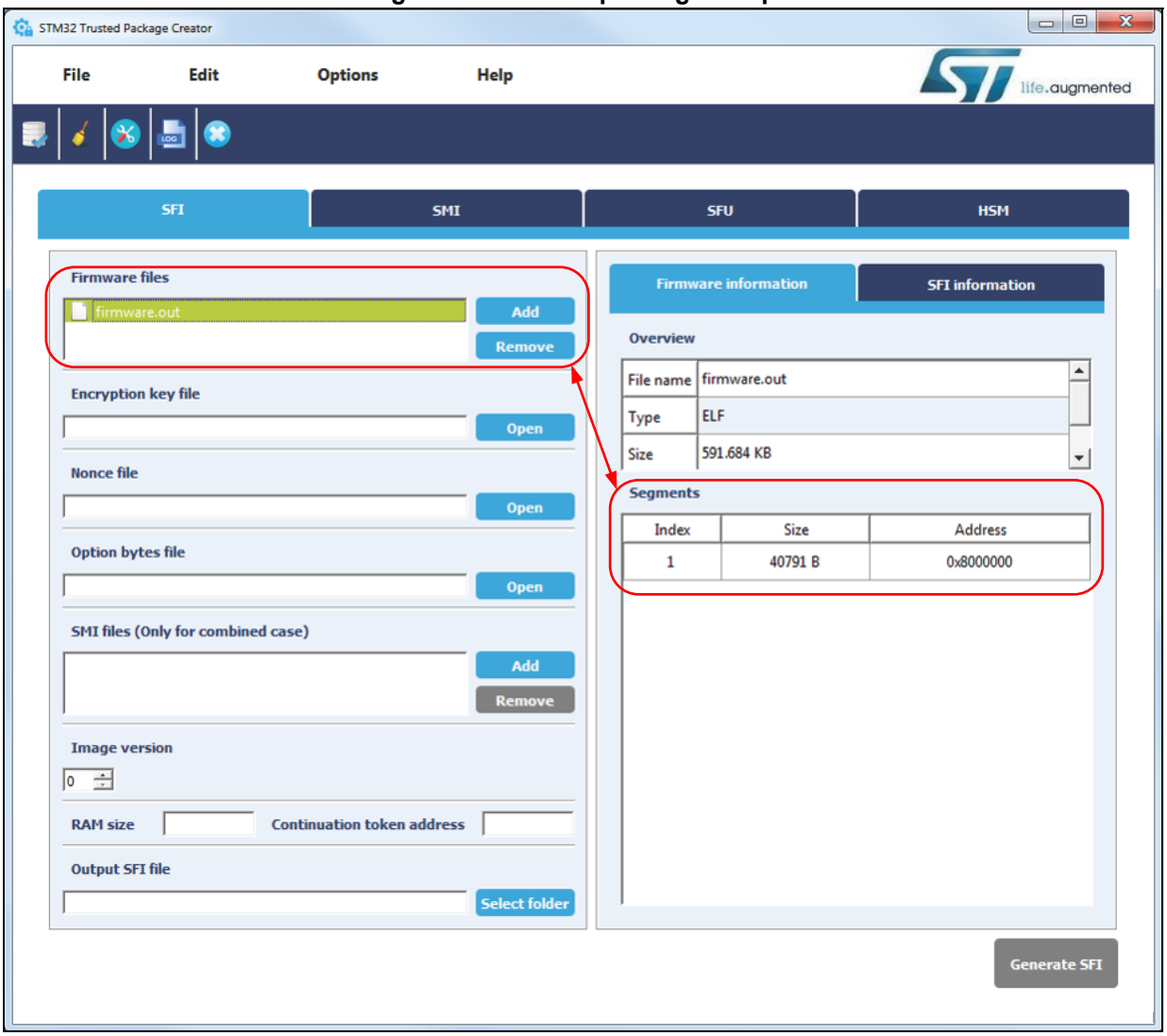


To successfully generate an SFI image from the supported input firmware formats, click the SFI GUI tab and fill in the interface fields with valid values:

1. Add the firmware file using the **Add** button located within the **Firmware files** area appended to the input firmware files list.
2. Make sure you are referring to the right valid firmware. Details are available in the **Firmware information** section once you have the Input firmware file is selected.

Note: STM32 Trusted Package Creator tool is included inside STM32CubeProgrammer tool package available on www.st.com.

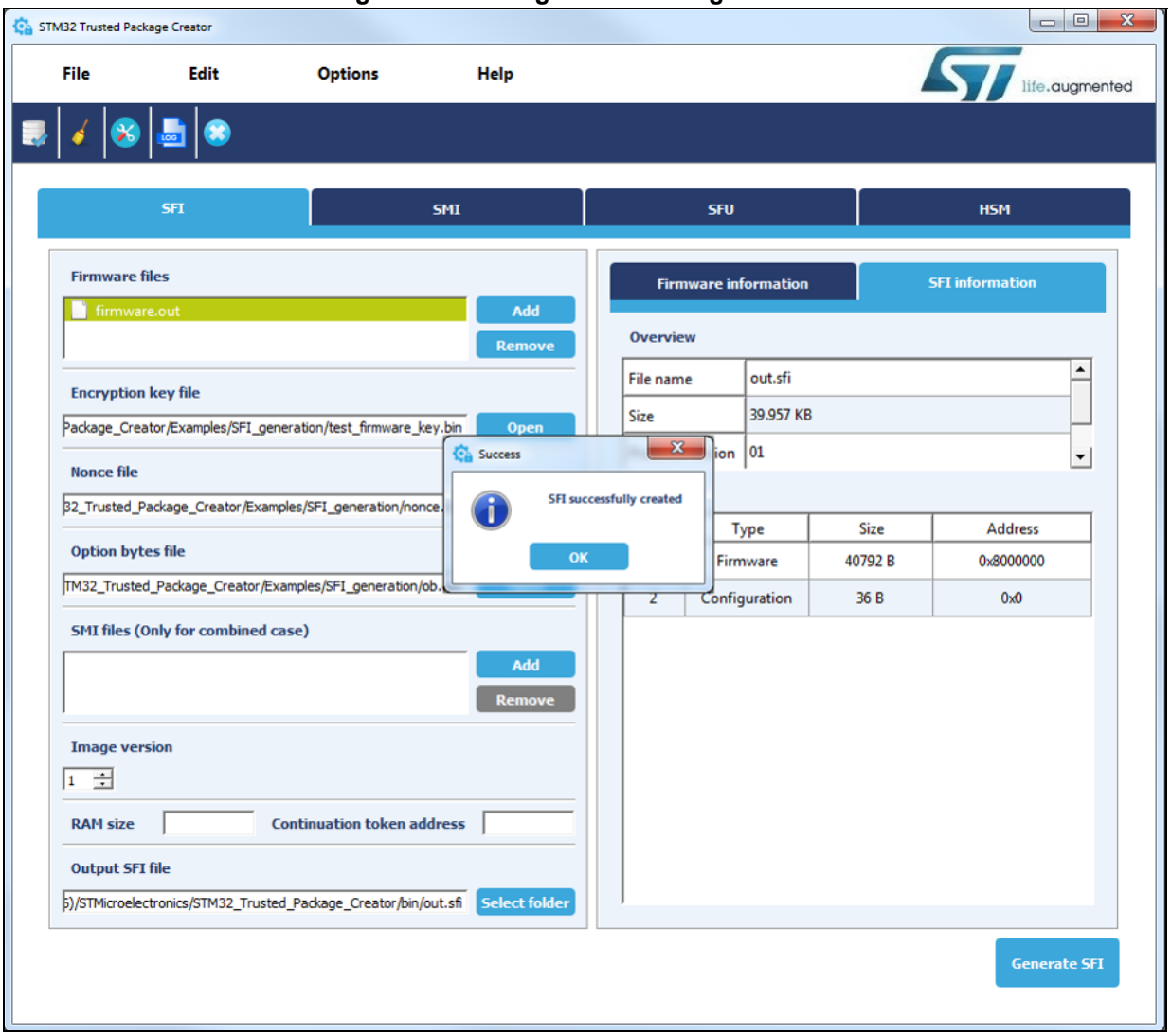
Figure 7. Firmware parsing example



3. Open the **Encryption key** and **Nonce** files. They can be selected either by entering their absolute or relative paths, or by selecting them with the **Open** button.
4. Make sure that the size of the Encryption key file (16 bytes) and the Nonce file (12 bytes) are respected.
5. Open the **Option bytes** file in .csv format. This is the only format supported.
6. Enter the image version of the SFI to be generated. It must range from 0 to 255.
7. Select the **Output** file folder path where the SFI image, *out.sfi*, is going to be generated.
8. Generate the SFI file by clicking the **Generate SFI** button. If all the fields are filled properly, the “SFI created successfully” message is displayed.

More information is available in *[UM2238]*.

Figure 8. SFI image successful generation



5 SFI HSM key provisioning

HSM smartcard[(a)] configuration is done using STM32 Trusted Package Creator tool following the below steps:

Note: HSM configuration can be done only once, the HSM is locked after successful programming.

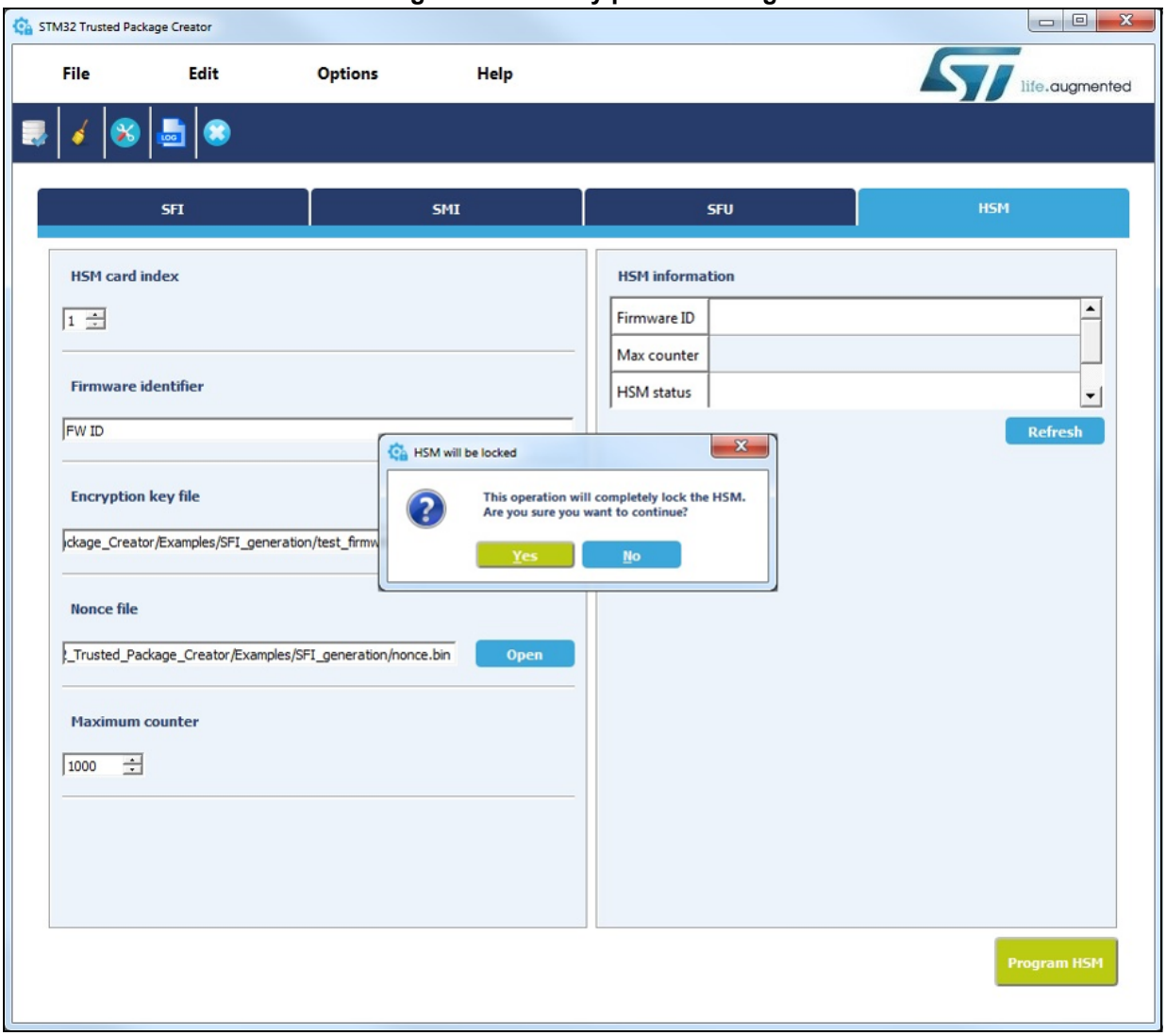
1. Insert a virgin HSM in smartcard reader.
2. Select **HSM** tab of STM32 Trusted Package Creator.
3. Add a firmware identifier (allows OEM to identify the correct HSM for a given firmware).
4. Open the **Encryption key** and **Nonce** files. They can be selected either by entering their absolute or relative paths, or by selecting them with the **Open** button.
5. Configure the maximum installation counter.
6. Configure HSM by clicking the **Program HSM** button. A feedback window indicating that HSM is going to be completely locked is displayed, to confirm and continue the procedure click **Yes**. If all the fields are filled properly, the “HSM programmed successfully” message is displayed.

After key provisioning, HSM can be used to secure install protected firmware on **counter** number of STM32 devices.

Note: More information is available in [UM2508].

a. Contact ST sales representative for HSM ordering information.

Figure 9. HSM key provisionning



6 SFI image programming by OEMs or CMs

6.1 Secure firmware installation flow

Once the SFI image has been prepared using the STM32 Trusted Package Creator, the OEM sends it to the CMs.

CM production lines need to be equipped with a Flash programming tool (FT). This tool is used to:

1. Download the SFI image prepared with STM32 trusted package creator, including the image header and the encrypted part of the SFI image.
2. For each STM32 device:
   a) Request the device certificate from the secure bootloader.
   b) Request a dedicated license from the HSM. The HSM uses the device certificate to produce a dedicated license for a given firmware to be stored in this particular STM32 device.
   c) Ask the secure bootloader to process the license, decrypt and flash the SFI image.

STM32CubeProgrammer is the Flash programming tool provided by STMicroelectronics. It is available in CLI mode and can be downloaded free of charge from *www.st.com*.

STM32CubeProgrammer can be used on OEM-CM production lines.

It supports the secure programming of SFI images for:

- STM32H753xI microcontrollers via USART, SPI, USB-DFU bootloader or JTAG interfaces.
- STM32L462CE microcontroller via USART or SPI interfaces.

STM32CubeProgrammer communicates with the chosen interface that triggers the secure bootloader in order to handle the OEMs' encrypted SFI image during SFI operations.

On STM32H753xI only, to start the SFI process, the bootloader activates the security via OB programming, which in turn activates the secure bootloader.

Note: More information on secure firmware installation sequence is available in:

- *[AN5243] for STM32H753xI*
- *[AN5251] for STM32L462CE*

STM32CubeProgrammer example command using USART interface on STM32 devices supporting SFI and with HSM smartcard usage:

- sfi command allowing secure installing of firmware "data.sfi" into STM32 user Flash memory:

```
STM32_Programmer_CLI.exe -c port=COM1 -sfi protocol=static
"C:\SFI\data.sfi" hsm=1 slot=1
```

More information is available in *[UM2237]* and *[AN5054]*.

7 Known limitations

7.1 STM32H753xI known limitations

SFI limitation

During a SFI involving a firmware which has to be written on last word of a Flash memory bank, an error is raised by the secure bootloader and the firmware is not programmed.

Recommendation

If the firmware to program in internal flash memory exceeds the size of one memory bank, it must be splitted in 2 parts (from linker point of view) with the first part avoiding the last word of the bank.

7.2 STM32L462CE known limitations

On STM32L462CE, the installed firmware size must be a multiple of 4 bytes and if greater than 64 Kbytes it must not be a multiple of 256 bytes to avoid any installation issue (padding bytes must be added in the binary to take into account these limitations).

8 Revision history

Table 3. Document revision history

Date	Revision	Changes
20-Dec-2018	1	Initial release.
12-Mar-2019	2	Added *Table 1: Applicable products* with indication of the STM32L462CEU6F (special order) code.
11-Jun-2019	3	Document is declassified.
11-Sep-2019	4	Restored and updated missing *Figure 2: STM32H753xI secure bootloader*.
07-Oct-2019	5	Updated *Table 2: Glossary terms*. Updated *Section 3.2.1: Secure bootloader overview*. Updated *Section 3.2.2: User Flash memory mapping*. Added *Section 7: Known limitations*.