

STM32 Trusted Package Creator tool software description

## Introduction

information.

STM32 Trusted Package Creator is part of the STM32CubeProgrammer tool set (STM32CUBEPROG), and allows the generation of secure firmware and modules to be used for STM32 secure programming solutions which are:

- Secure firmware install (SFI): SFI is a secure mechanism that allows secure installation
  of OEM firmware in untrusted production environments by having the whole firmware
  encrypted with an AES-GCM key.
- Secure module install (SMI): SMI is intended to protect a part of the firmware (a section of an ELF file) by also encrypting this section using an AES-GCM key.
   A combined SFI-SMI image is an SFI image that contains one or more module areas.
- Secure firmware upgrade (SFU): SFU is a solution to allow the upgrade of the STM32 microcontroller built-in program in a secure way. For more information about SFU, please refer to the X-CUBE-SBSFU package on http://www.st.com for further

This user manual details the software environment prerequisites, as well as the available features of the STM32 Trusted Package Creator tool software.





DocID030724 Rev 1

1/33

# Contents

1	Syst	em requirements
2	Prep	paration processes
	2.1	SFI preparation process
	2.2	SMI preparation process
	2.3	SFU preparation process 10
3	STM	32 Trusted Package Creator tool commands
	3.1	Command line interface (CLI) 12
	3.2	SFI generation command
	3.3	SMI generation command
	3.4	SFU generation command 16
4	STM	32 Trusted Package Creator tool graphical user interface (GUI) 18
	4.1	SFI generation
	4.2	SMI generation
	4.3	SFU generation
5	Opti	on bytes file
6	Log	dialog
7	Sett	ings
8	SFI/S	SMI checking
9	Revi	sion history



# List of tables

Table 1	Decument revision history	22
Table T.	Document revision history	 JZ



# List of figures

Figure 1.	SFI preparation process	6
Figure 2.	SFI file structure	7
Figure 3.	SMI preparation process.	8
Figure 4.	SMI file structure	9
Figure 5.	SFU preparation process 1	0
Figure 6.	SFU file structure	1
Figure 7.	STM32 Trusted Package Creator tool's available commands 1	2
Figure 8.	SFI generation with an ELF file 1	4
Figure 9.	SFI generation with a binary file 1	4
Figure 10.	Combined SFI-SMI generation 1	4
Figure 11.	SMI generation	6
Figure 12.	SFU generation	7
Figure 13.	STM32 Trusted Package Creator tool GUI SFI tab1	8
Figure 14.	STM32 Trusted Package Creator tool GUI SMI tab 1	9
Figure 15.	STM32 Trusted Package Creator tool GUI SFU tab	20
Figure 16.	Firmware file addition	!1
Figure 17.	Successful SFI generation	2'
Figure 18.	ELF file selection	23
Figure 19.	Successful SMI generation	25
Figure 20.	Firmware file selection	26
Figure 21.	Successful SFU generation	27
Figure 22.	Example of an option bytes file	28
Figure 23.	Example of a log dialog	29
Figure 24.	Settings dialog	0
Figure 25.	SFI checking	51



# 1 System requirements

Supported operating systems and architectures:

- Linux<sup>®</sup> 32-bit and 64-bit (tested on Ubuntu 14.04)
- Windows<sup>®</sup> 7/8/10 32-bit and 64-bit
- macOS<sup>®</sup> (minimum version OS X<sup>®</sup> Yosemite)

STM32CubeProgrammer and STM32 Trusted Package Creator support STM32 32-bit devices based on  $\rm Arm^{\$}$  Cortex  $^{\$}$ -M processors.





## 2 Preparation processes

## 2.1 SFI preparation process

An SFI (Secure firmware install) image is a format created by STMicroelectronics that contains an encrypted and authenticated piece of firmware using an AES-GCM algorithm. The SFI preparation process is described in *Figure 1*.



Figure 1. SFI preparation process

DocID030724 Rev 1



Before performing AES-GCM to encrypt an area, the tool calculates the Initialization Vector (IV) as:

Where nonce is a number used only once as a start of an iterated process in the AES-GCM algorithm to give different cipher texts to same blocks of data.

It then, it passes the area descriptor (starting from the magic to the total number of areas) as additional authenticated data (AAD).

- Each segment in the input firmwares constitutes a firmware (F) area in the SFI file.
- Each SMI file (combined case) constitutes a module (M) area.
- The option bytes configuration constitute the configuration (C) area.

To generate a header tag, the tool performs an authenticated only AES-GCM encryption (without a plain text nor a cipher text) using the SFI header as an AAD and the nonce as the IV.

The structure of an SFI file is shown in *Figure 2*.







To prepare an SFI image from multiple firmware files you have to make sure that there is no overlap between their segments, otherwise you will get the error message: *"Overlap between segments, unable to merge firmware files"*.

Furthermore, in the case of a combined SFI-SMI image, there is also an overlap check between the areas (in case there is an overlap between firmware and module areas). If the check fails, an error message is shown: *"Overlap between SFI areas"*.

Also, all SFI areas must be located in Flash memory, otherwise the generation fails gving the error message: *"One or more SFI areas are not located in Flash memory"*.

## 2.2 SMI preparation process

An SMI image (Secure Module Install) protects only a module of the firmware.

The SMI preparation process is shown in *Figure 3*.





The AES-GCM encryption is performed using the following inputs:

- Nonce as the initialization vector (IV)
- The security version as additional authenticated Data (AAD)



Before SMI preparation, the following checks are performed:

- A proprietary code read out protection (PCROP) section must be aligned on a Flash word (256 bits) otherwise a warning is shown
- The section's size must be at least 2 Flash words (512 bits) otherwise a warning is shown
- The section must end on a Flash word boundary (a 256 bit word) or a warning is shown
- If the section immediately following the PCROP area starts in the last Flash word of the PCROP section, the generation fails and an error message is shown.

After the SMI preparation, a clear (that is, not encrypted) ELF file is also generated containing program data and only clear sections of the code.

The structure of an SMI file is shown in *Figure 4*.







## 2.3 SFU preparation process

An SFU image (Secure Firmware Upgrade) allows the upgrade of the STM32 microcontroller built-in program in a secure way to prevent unauthorized updates.

The SFU preparation process is shown in *Figure 5*.







2 files are generated, the SFU image header and the SFU encrypted firmware image.

For the header generation, the AES-GCM encryption is performed using the following inputs:

- Nonce as the Initialization Vector (IV)
- No Additional Authenticated Data (AAD)
- And for the encrypted firmware image generation:
- Nonce as the Initialization Vector (IV)
- The header content as Additional Authenticated Data (AAD)

The structure of an SFU file is shown in *Figure* 6.







# **3 STM32 Trusted Package Creator tool commands**

## 3.1 Command line interface (CLI)

The following sections describe how to use the STM32 Trusted Package Creator tool from the command line interface.

The available commands are shown in Figure 7

Figure 7. STM32 Trusted Package Creator tool's available commands

	STM32TrustedPackageCreator v1.0
lsage : SFMIPreparationTool_CLI	.exe [option1] [value1] [option2] [value2]
Information options	
-?, -h,help -l,log [ <file_name>]</file_name>	: Display this help : Generate a log file : Log file's path and name, default file is ./trace.log
SFI preparation options	
-sfi,sfi -fir,firmware <firm_file> [<address>] -k,key <fiestkey <fiestkey <fiestkey -k,key <fiestkey <fiestkey -k,key key st</fiestkey </fiestkey </fiestkey </fiestkey </fiestkey </address></firm_file>	<pre>: Generate SFI image,</pre>
MI preparation options	
-smi,smi, -elf,elfile <elf_file> -s,sec <section> -k,key <key_file> -n,nonce <nonce_file> -sv,sver <su_file> -o,outfile -c,clear <clear_file> SFU prenaration options</clear_file></su_file></nonce_file></key_file></section></elf_file>	<pre>: Generate SMI image You also need to provide the information listed below Input ELF file : ELF file : Section to be encrypted : Section name in the ELF file : AES-GCM encryption key Bin file, its size must be 16 bytes : AES-GCM nonce : Bin file, its size must be 12 bytes : Security version : Its size must be 16 bytes : Generated SMI file : SMI file to be created : Clear ELF file : Clear ELF file to be generated</pre>
-sfusfu	· Cenevate SEU image
-fir,firmware {Firm_File> [ <address>] -k,key <key_file> -n,nonce</key_file></address>	You also need to provide the information listed below Add an input firmware file (must have only 1 segment) Supported firmware files are ELF HEX SREC BIN Only in case of BIN input file (in any base) AES-GCM encryption key Bin file, its size must be 16 bytes AES-GCM nonce Bin file, its size must be 12 bytes
<pre></pre>	: Image version : Its value must be in <0255> (in any base) : Generated SFU header file

DocID030724 Rev 1



## 3.2 SFI generation command

-sfi, --sfi

Description: This command generates an SFI image file.

In order to generate an SFI image, the user must provide the mandatory inputs by using the options listed below.

-fir, --firmware

**Description**: Add an input firmware file. Supported formats are Bin, Hex, Srec and ELF. This option can be used more than once in order to add multiple firmware files.

Syntax: -fir <Firmware\_file> [<Address>]

<Firmware\_file> : Firmware file.

[<Address>] : Address only for binary firmware.

-k, --key

**Description**: Set the AES-GCM encryption key.

Syntax:	-k	<key_file></key_file>
---------	----	-----------------------

< Key \_file> : A 16-byte binary file.

-n, --nonce

Description: Set the AES-GCM nonce.

Syntax: -n	<nonce_file></nonce_file>
------------	---------------------------

<Nonce\_file> : A 12-byte binary file.

-v, --ver

**Description**: Set the image version.

**Syntax**: -v <Image\_version>

<Image\_version> : A value between 0 and 255 in any base.

-ob, --obfile

Description: Provide an option bytes file.

Syntax: -ob <CSV\_file>

<CSV\_file>: A csv file with 9 values.

#### -m, --module

**Description**: Add an input SMI file. This option can be used more than once in order to add multiple SMI files. This is optional (for combined SFI-SMI).

Syntax: -m <SMI\_file> <SMI\_file>

[<Address>] : Address only for relocatable SMI.

#### -o, --outfile

Description: Set the SFI file to be created



Syntax: -o <out\_file>

<out\_file>: SFI file to be generated, must have the .sfi extension.

Example:

With an ELF file:

```
STM32TrustedPackageCreator_CLI -sfi -fir ELF_firmware.axf
-k
test_firmware_key.bin -n nonce.bin -ob FIR_ob.csv -v
23 -o test.sfi
```

Figure 8. SFI generation with an ELF file



With a binary file:

STM32TrustedPackageCreator\_CLI -sfi -fir bin\_firmware.bin 0x8000000 -k test\_firmware\_key.bin -n nonce.bin -ob FIR\_ob.csv -v 23 -o test.sfi

#### Figure 9. SFI generation with a binary file

C:\STM32TrustedPackageCreator\bin>STM32TrustedPackageCreator\_CLI.exe -sfi -fir b in\_firmware.bin 0x08000000 -k test\_firmware\_key.bin -n nonce.bin -ob FIR\_ob.csv -v 23 -o test.sfi SUCCES

#### Combined SFI-SMI:

STM32TrustedPa	ackag	eCreator_CLI	-sfi	-fir	ELF_fir	mwrae	.axf	-fir
bin_firmware.k	oin	0x8000000	-m	FIR_pc	rop.smi	-k	test	_firm-
ware_key.bin	-n	nonce.bin	-ob	FIR_	ob.csv	-v	23	-0
test.sfi								

#### Figure 10. Combined SFI-SMI generation

C:\STM32TrustedPackageCreator\bin>STM32TrustedPackageCreator\_CLI.exe -sfi -fir b in\_firmware.bin 0x08000000 -m FIR\_pcrop.smi -k test\_firmware\_key.bin -n nonce.bi n -ob FIR\_ob.csv -v 23 -o test.sfi SUCCES



### 3.3 SMI generation command

#### -smi, --smi

Description: This command generates an SMI image file.

In order to generate an SMI image, the user must provide the mandatory inputs by using the options listed below.

-elf, --elfile

Description: Set the input ELF file.

```
Syntax: -elf <ELF_file>
```

<ELF\_file>

: ELF file. An ELF file can have any of the extensions: .elf, axf, .o, so, .out

#### -s, --sec

Description: Set the name of the section to be encrypted.

Syntax: -s	<section_< th=""><th>_name&gt;</th></section_<>	_name>
------------	---	--------

<section\_name>: Section name.

-k, --key

**Description**: Set the AES-GCM encryption key.

Syntax: -k <Key\_file>

<Key \_file> : A 16-byte binary file.

-n, --nonce

Description: Set the AES-GCM nonce.

```
Syntax: -n <Nonce_file>
```

<Nonce \_file>: A 12-byte binary file.

#### -sv, --sver

Description: Set the security version file.

```
Syntax: -sv <SV_file>
```

<SV\_file>: A 16 byte file.

#### -o, --outfile

Description: Set the SMI file to be created

**Syntax**: -o <out\_file>

<out\_file>: SMI file to be generated, must have the .smi extension.

-c, --clear

Description: Set the clear ELF file to be created.

**Syntax**: -c <ELF\_file>

<ELF\_file>: Clear ELF file to be generated.



#### Example

STM32TrustedPackageCreator\_CLI -smi -elf FIR\_module.axf -s "ER\_PCROP" -k test\_firmware\_key.bin -n nonce.bin -sv svFile -o test.smi -c clear.smi

#### Figure 11. SMI generation

```
C:\STM32TrustedPackageCreator\bin>STM32TrustedPackageCreator_CLI.exe -smi -elf F
IR_module.axf -s "ER_PCROP" -k test_firmware_key.bin -n nonce.bin -sv svFile -o
test.smi -c clear.axf
Warning: The section does not end on a Flash word boundary
SUCCES
```

## 3.4 SFU generation command

#### -sfu, --sfu

Description: This command generates an SFU image file.

In order to generate an SFU image, the user must provide the mandatory inputs by using the options listed below.

#### -fir, --firmware

**Description**: Set an input firmware file. Supported formats are Bin, Hex, Srec and ELF. The firmware file must contain only one segment.

**Syntax**: -fir <Firmware\_file> [<Address>]

< Firmware \_file>: Firmware file.

[<Address>]: Address only for binary firmwares.

-k, --key

**Description**: Set the AES-GCM encryption key.

Syntax: -k <Key\_file>

<Key \_file>: A 16-byte binary file

-n, --nonce

**Description**: Set the AES-GCM nonce.

Syntax: -n <Nonce\_file>

<Nonce \_file>: A 12-byte binary file.

-v, --ver

**Description**: Set the image version.

**Syntax**: -v <Image\_version>

<Image\_version>: A value between 0 and 255 in any base.

-oh, --outheader

Description: Set the SFU header file to be created.

Syntax: -oh <out\_file>



<out\_file>:

SFU header file to be generated, must have the .sfuh extension.

-os, --outsfu

Description: Set the SFU file to be created.

Syntax: -os <out\_file>

<out\_file>: SFU file to be generated, must have the .sfu extension.

Example:

SFMIPreparationTool\_CLI -sfu -fir bin\_firmware.bin -k test\_firmware\_key.bin -n nonce.bin -v 23 -oh out.sfuh -os out.sfu

#### Figure 12. SFU generation

C:\SIM32TrustedPackageCreator\bin>SIM32TrustedPackageCreator\_CLI.exe -fir bin\_fi rmware.bin -k test\_firmware\_key.bin -n nonce.bin -v 23 -oh out.sfuh -os out.sfu -sfu SUCCES



# 4 STM32 Trusted Package Creator tool graphical user interface (GUI)

This section describes how to use the STM32 Trusted Package Creator tool with its graphical user interface.

The STM32 Trusted Package Creator tool GUI presents three tabs: one for SFI generation (*Figure 13*), one for SMI generation (*Figure 14*) and one for SFU generation (*Figure 15*).

	- u.	<b>.</b>				
File	Edit	Options	Help			life.augment
🤞 💰	) 🔙 😣					
	SFI		SMI			SFU
		_		_	_	
Firmware	files		Add	Firmware i	nformation	SFI information
			Remove	Overview		
Encryptio	n key file			File name		
			Open	Туре		
Nonce file				Size		
			Open	Segments		
Option by	tes file			Index	Size	Address
			Open			
SMI files (	(Only for combined	case)				
			Add Remove			
Image ve	rsion					
0						
Output SF	I file					
			Select folder			
						Generate SFI

Fiaure 13.	STM32	Trusted	Package	Creator too	GUI SFI tab
	0111102	1140104	i uonugo	0104101 100	00101100



STM32TrustedPackageCreator         File       Edit       Options       Help         SFI       SHI       STU         SFI       SHI       STU         ELF file       Open       SHI information       SHI information         Encryption key file       Open       Open       SHI information         Nonce file       Open       Security version file       Open         Security version file       Open       Sections       Index         Output SHI file       Open       Select folder       Index       Name       Type       Size         Output clear ELF file       Select folder       Index       Index       Index       Index       Index						gu	
File Edit Options Help     Image: Spin and spi		2				PackageCreator	STM32Trusted
ST SHI     ELF file     Open     Encryption key file     Open     Nonce file     Open     Security version file     Open     Section to encrypt     Output SHI file     Section to encrypt     Output Cear ELF file     Sect folder     Output cear ELF file	e.augmented	life.c		Help	Options	Edit	File
SFI     SHI     SFU       ELF file     Open     Overview       Encryption key file     Open     Overview       Nonce file     Open     Security version file     Size       Security version file     Open     Sections       Section to encrypt     Index     Name     Type       Output SHI file     Select folder     Index     Index						3 🔤 🕄	J 🤞
ELF file   Open   Encryption key file   Open   Nonce file   Open   Security version file   Open   Section to encrypt   Output SMI file   Output SMI file   Output Clear ELF file   Select folder		SFU		SMI		SFI	
Open   Encryption key file   Open   Open   Nonce file   Open   Security version file   Open   Section to encrypt   Output SMI file   Output Clear ELF file   Select folder	n	SMI information	ELF information				ELF file
Encryption key file   Open   Nonce file   Open   Security version file   Open   Section to encrypt   Output SMI file   Output clear ELF file   Select folder			Overview	Open			
Open   Nonce file   Open   Security version file   Open   Section to encrypt   Output SMI file   Output clear ELF file   Select folder			File name			on key file	Encrypti
Nonce file   Open   Security version file   Open   Section to encrypt   Output SMI file   Output clear ELF file   Select folder			ELF Type	Open			
Open Size   Security version file   Open Sections     Open Sections     Output SMI file     Output Clear ELF file     Select folder			ELF Machine			e	Nonce fil
Security version file       Open       Section to encrypt       Output SMI file       Output clear ELF file       Select folder			Size	Open			
Section to encrypt		Type Size	Index Name	Ореп		version file	Security
Output SMI file     Output clear ELF file     Select folder     III						to encrypt	Section
Output SMI file  Output clear ELF file  Select folder						•	
Select folder       Output clear ELF file       Select folder						MI file	Output S
Output clear ELF file        Select folder				Select folder			
Select folder						lear ELF file	Output c
	4		< <u> </u>	Select folder			
Gen	erate SMI	Genera					

#### Figure 14. STM32 Trusted Package Creator tool GUI SMI tab



File	Edit	Options	Help	life.augment
6 💰	) 📠 😣			
	, , 		CHI	CEIL
	511		311	
Firmware	file			Firmware information
			Open	File name
				Туре
Encryptio	n key file			File size
			Open	Segment address
				Segment size
Nonce file	2			SFU information
			Open	SFU header file name
Image ve	reion			SFU file name
				Size of encrypted area
0 ਵ				Protcol version
Output Si	FU header file			Image version
			Select folder	
Output Si	FU file			
			Select folder	

#### Figure 15. STM32 Trusted Package Creator tool GUI SFU tab

## 4.1 SFI generation

To validate the SFI generation request, the user has to fill in the input fields with valid values:

#### Firmware files:

The user mst add the input firmware files with the **add** button.

Note: If the file is valid, it is added in the firmware files list. Selecting it makes several pieces of related information it appear in the Firmware information section (Figure 16), otherwise an error message box is shown saying either the file could not be opened or the file is not valid.

If the file is in a binary format, a dialog box appears requesting that an address be provided. A file can be removed with the **remove** button.



File	Edit	Options	Help			
🤞 😵		-	·			
	SFI		SMI			SFU
Firmware	files mware.axf		Add	Firmwa	re information	SFI information
			Remove	Overview		
Encryption	n key file			File name	ELF_firmware.axf	
			Open	Туре В	ELF	
Nonce file				Size 8	815.887 KB	
			Open	Segments		
Option by	tes file			Index	Size	Address
			Open	1	844 B	0x8000000
CMT files /	Only for combined	(2562)		2	9884 B	0x8030000
			Add Remove			
Image ve	rsion					
Output SF	I file					
			Select folder			
						Conorato SEL

Figure 16. Firmware file addition

#### Encryption key and nonce file:

The encryption key and nonce file can be selected by entering their paths (absolute or relative), or by selection with the **open** button. Notice that sizes must be respected (16 bytes for the key and 12 bytes for nonce).

#### Option bytes file:

The option bytes file can be selected the same way. Only csv files are supported.

#### SMI files:

SMI files can be added the same way as the firmware files. Selecting a file makes several pieces of related information appear in the Firmware information section.

#### Image version:

Image version value in [0..255].



DocID030724 Rev 1

#### Output file:

An output file can be selected by entering its path (absolute or relative), or with the **select folder** button, note that with the latter way, a name *out.sfi* is suggested, you can keep or change it

When all fields are properly filled in, the **Generate SFI** button becomes active. The user may generate the SFI file by clicking on it.

If everything goes well, a message box indicating successful generation appears (*Figure 17*) and information about the generated SFI file is displayed in the SFI information section.

🤞 🛛 🍪								
	SFI		SM	I			5	FU
Firmware f	iles				irmware	information	51	-I information
ELF_firm	nware.axf		Add Remove	Ove	rview			
Encryption	key file			File	name	out.sfi		
C:/STM32Trus	tedPackageCreator/l	bin/test_firmware_key.	bin Open	Size		10.627 KB		
Nonce file		(***		Prot	ocol versi	ion 01		
C:/STM32Trus	tedPackageCreator/	ti Success		Seg	ments			
Ontion byte	es file	SFI	successfully created	Ind	ex	Туре	Size	Address
C. /CTM20Teur	todDadvageCreater/			1		Firmware	844 B	0x8000000
C:/STM521108	aleurackagecreator/		ок	2		Firmware	9884 B	0x8030000
SMI files (0	only for combined	ca,		3	Co	onfiguration	36 B	0x0
			Add Remove					
Image ver	sion							
23 🚖								
Output SFI	file							
C:/STM32Trus	stedPackageCreator/	bin/out.sfi	Select folder					
:/STM32Trus	stedPackageCreator/l	bin/out.sfi	Select folder					$\sim$

Figure 17. Successful SFI generation



## 4.2 SMI generation

As for SFI generation, the user must provide the input information.

#### Elf file:

In this case the input file can be only an elf file.

If the file is valid, information is displayed in the *"ELF information"* tab (*Figure 18*), otherwise an error message box is shown saying either the file could not be opened or that the file is not valid.

File	Edit	Options	Help				1ife.aug
🤞 😵	5						
	SFI		SMI			SF	U
ELF file				ELF	information	SMI	information
C:/STM32Tru	stedPackageCreator/	pin/FIR_module.axf	Open	Overview			
Encryption	ı key file			File name	FIR_module.a	xf	
			Open	ELF Type	Executable		
Nonce file				ELF Machin	e ARM		
			Open	Size	14.1328 KB		
Security y	ersion file			Sections			
Jecuncy -			Open	Index	Name	Type NULL	Size U B
				2	ER_PCROP	PROGBITS	2484 B
Section to	encrypt			3	ER_PRDATA	PROGBITS	956 B
EK_PCKOP	•			4	RW_IRAM1	NOBITS	240 B
Output SM	I file			5	.note	NOTE	28 B
			Select folder	6	.comment	PROGBITS	3316 B
Output cle	ar ELF file			7	.shstrtab	STRTAB	56 B
			Select folder	•	111		4
							Generate
							Generate

Figure 18. ELF file selection



#### Encryption key and nonce file:

As for SFI, the Encryption key and nonce file can be selected in the same way as the firmware file. Notice that sizes must be respected (16 bytes for the key and 12 bytes for nonce).

#### Security version file:

The security version file size must be 16 bytes.

A security version file is provided under the Security\_Version folder.

#### Section:

This is a section list that can be used to select the name of the section to be encrypted.

#### output files:

Output files can be selected by entering their paths (absolute or relative), or with the **Select folder** button, note that with the latter way a name is suggested, which can be kept or changed.

When all fields are properly filled in, the user may generate the SFI file by a click on the **Generate SFI** button (the button becomes active).

A message box informing the user that generation was successful appears (*Figure 19*), in addition of information about the generated SMI file, otherwise an error is displayed.



STM32TrustedP	'ackageCreator							
File	Edit	Options	Help				57	life.augmen
🤞 💰	ا 🕹							
	SFI			БМІ			SFU	
ELF file					ELF informa	ation	SMI informa	ation
C:/STM32Tru	.stedPackageCreator/bi	n/FIR_module.axf	Open		Overview			
Encryption	ı key file				Original file name	FIR_module.smi		
C:/STM32Tru	.stedPackageCreator/bi	in/test_firmware_key.bin	Open		Number of files	1		
Nonce file					Last file size	2.47266 KB		
C:/STM32Tru	istedPackageCreator/bi	in/nonce.bin	Open		SMI address	0x8080000		
Security v	ersion file	¢.	Information					
C:/STM32Tru	istedPackageCreator/bi	in/svFile	SMI suc	cessful	ly created			
Section to	encrypt				ок			
ER_PCROP	•			_				
Output SM	II file							
C:/STM32Tru	.stedPackageCreator/bi	in/FIR_module.smi	Select folde					
Output cle	ar ELF file							
C:/STM32Tru	istedPackageCreator/bi	in/FIR_module_clear.axf	Select folde					-
								Generate SMI

#### Figure 19. Successful SMI generation

## 4.3 SFU generation

Input fields in this case are similar to the SFI generation use case.

#### Firmwares files:

The user needs to enter the input firmware files.

If the file is valid, several pieces of related information appear in the Firmware information section (*Figure 20*), otherwise an error message box is shown. Note that the firmware file must have only one segment.



1	1 1		•		life.	augment
🤞 😵						
	657		CMT		CE11	
	SFI		5111		310	
Firmware	file			Firmware inform	nation	
C:/STM32Tri	istedPackageCreator/	/bin/file.bex	Open	File name	file.hex	
enjo mozini	astear actageor catory	onymentex	open	Туре	Intel Hex	
Encryptio	n key file			File size	27.2002 KB	
			Open	Segment address	0x8030000	
				Segment size	9884 B	
Nonce file				SFU information	I.	
			Open	SFU header file na	ame	
Image ve	rsion			SFU file name		
				Size of encrypted a	area	
0 🖵				Protcol version		
Output SF	U header file			Image version		
			Select folder			
Output SF	U file					
			Select folder			
						_

#### Figure 20. Firmware file selection

#### Encryption key and nonce file:

Same as for SFI and SMI use cases.

#### Image version:

Image version value in [0..255].

#### output files:

Output files can be selected by entering their paths (absolute or relative), or with the **select folder** button, note that in the latter way a name is suggested, which can be kept or changed.

When all fields are properly filled in, the **Generate SFU** button becomes active. The user may generate the SFU file by clicking on it.

If everything goes well, a message box informing that generation is successful appears (*Figure 21*) and information about the generated SFU file is displayed in the SFU information section.

DocID030724 Rev 1



M32TrustedP	ackageCreator					
File	Edit	Options	Help			life.augme
🤞 🕉	5					
	SFI		SMI			SFU
Firmware f	ĭle			Firmware inform	ation	
0. (07140.07		- /Cl- 1		File name	file.hex	
C:/STM32Tru	stedPackageCreator/d	bin/file.nex	Open	Туре	Intel Hex	
Encryption	key file			File size	27.2002 KB	
C:/STM32Tru	stedPackageCreator/b	pin/test_firmware_key.bin	Open	Segment address	0x8030000	
				Segment size	9884 B	
Nonce file				SFU information		
C:/STM32Tru	stedPackageCreator/b	pin/nonce.bin	Open	SFU header file na	me file.sfuh	
-		Ca Success	×	SFU file name	file.sfu	
Image ver	SION	SFU S	successfully created	Size of encrypted	area 9.65234 KB	
7 🚖				Protcol version	1	
Output SFL	J header file		ок	Image version	7	
C:/STM32Tru	stedPackageCreator/b	pin/file.sfuh	Select folder			
Output SFL	J file					
C:/STM32Tru	stedPackageCreator/b	pin/file.sfu	Select folder			
						Generate SF

Figure 21. Successful SFU generation



# 5 Option bytes file

The option bytes file field is mandatory for SFI applications only, it allows option bytes to be programmed during secure firmware install.

Only CSV (Comma Separated Value) format is supported for such files, it is composed of two vectors: a register name and its value.

All of the 9 option-byte registers must be configured (a total of 9 lines in a csv file).

🔚 FIR_c	ob.csv 🔀
1	FOPTSR_PRG, 0x13FE00F8
2	FPRAR_PRG_A,0x88060800
3	FPRAR_PRG_B,0x80000FFF
4	FSCAR_PRG_A,0x80000FFF
5	FSCAR_PRG_B,0x80000FFF
6	FWPSN_PRG_A, 0x000000FF
7	FWPSN_PRG_B,0x000000FF
8	FBOOT7_PRG, 0x1FF10800
9	RFU,0x10000810

Figure 22. Example of an option bytes file



# 6 Log dialog

A log can be visualized by clicking the **log** button in the tools bar or in the menu bar: Options-> log.

STM32TrustedP	ackageCreator				
File	Edit	Options H	lelp		
💂 🤞 😵					
	SFI	Cog		? ×	SFU
Firmware f	file	17:24:43:168 SFI prepa 17:24:43:168 Area 1 pr 17:24:43:168 Area 2 pr 17:24:43:168 Area 3 pr 17:24:43:168 SFI head 17:24:43:168 SFI prepa	aration started 'epared with size 844 : firmware area epared with size 9884 : firmware are 'epared with size 36 : option bytes ar er prepared vration finished	ea rea	
C:/STM32Tru	istedPackageCreator/bi	n/file. 17:26:15:635 SMI prepa 17:26:15:635 1 SMI to p 17:26:15:635 SMI data	aration started prepare prepared with size 2484		
Encryption	ı key file	17:26:15:635 SMI head 17:26:15:635 SMI prepa 17:28:48:336 SFU prepa 17:28:48:336 Encrute	ler prepared aration finished aration started d image prepared with size 9884		
C:/STM32Tru	istedPackageCreator/bi	n/test 17:28:48:336 SFU head 17:28:48:336 SFU prep	ler prepared aration finished		
Nonce file	-to dDe due of Country (hi				
Image ver	scion				
7	51011			КВ	
Output SFL	U header file				
C:/STM32Tru	stedPackageCreator/bi	n/file.			
Output SFU	U file				
C:/STM32Tru	istedPackageCreator/bi	n/file.sfu	Select folder		
					Generate SFU

Figure 23. Example of a log dialog



# 7 Settings

The settings dialog can be accessed by clicking the **settings** button in the tools bar or in the menu bar: Options -> settings.

File	Edit	Options	Help			
1	<b></b>					
	SFI		SMI			SFU
Firmware	file		Settings	? ×		
C:/STM32Tru	ustedPackageCreator/	bin/file.hex	Padding byte		hex	
Encryption	n key file		<ul> <li>0x00</li> <li>0xFF</li> </ul>		002 KB	
C:/STM32Tru	ustedPackageCreator/	bin/test_firmware	Settings file		)30000	
Nonce file			☑ Generate settings file		4 B	
C:/STM32Tru	ustedPackageCreator/	bin/nonce.bin	Log file		file.sfuh	
Image ver	rsion		Generate log file	Select folder	file.sfu 9.65234 KB	
7 荣			ок	Cancel	1	
C:/STM32Tru	J header file ustedPackageCreator/	/bin/file.sfuh	Select folder		J ′	
Output SF	U file					
C:/STM32Tru	ustedPackageCreator/	bin/file.sfu	Select folder			
						Concepto 550

Figure 24. Settings dialog

#### Padding byte:

When parsing files, padding may be added to fill the gap between segments separated by 16 bytes or less, in order to merge them and reduce the number of segments. The user may have the choice between 0xFF (default value) or 0x00.

#### Settings file:

When checked, a *settings.ini* file is generated in the executable folder. It saves the application state: window size and field contents.

#### Log file:

When checked, a log file is generated in the selected path.

DocID030724 Rev 1



# 8 SFI/SMI checking

SFI/SMI checking can be accessed by clicking the **Check SFI/SMI** button in the tools bar or in the menu bar: File -> Check SFI/SMI.

This allows SFI or SMI file validity to be checked, in addition to displaying information about it.

File	Edit	Options	Help					life.augmen
1 🖉								
	SFI		SMI				5	FU
Firmware	files			Firn	nware info	rmation	SF	T information
			Remove	Overvi	ew			
Encryptio	n key file			File nam	ne	out.sfi		
		Si Si	uccess X	Size		10.627 KB		
Nonce file	:		Valid SFI file	Protoco Segme	l version nts	01		
				Index	Ту	pe	Size	Address
Option by	tes file		ок	1	Firm	ware	844 B	0x8000000
		_	Open	2	Firm	ware	9884 B	0x8030000
SMI files	(Only for combined o	case)		3	Configu	uration	36 B	0x0
			Remove					
Image ve	rsion							
Output St	T file							
output 5	* III.		Select folder					
								Generat <u>e SFI</u>

Figure 25. SFI checking



# 9 Revision history

Table 1.	Document	revision	history
----------	----------	----------	---------

Date	Revision	Changes
20-Dec-2017	1	Initial release.



#### IMPORTANT NOTICE - PLEASE READ CAREFULLY

STMicroelectronics NV and its subsidiaries ("ST") reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice. Purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST's terms and conditions of sale in place at the time of order acknowledgement.

Purchasers are solely responsible for the choice, selection, and use of ST products and ST assumes no liability for application assistance or the design of Purchasers' products.

No license, express or implied, to any intellectual property right is granted by ST herein.

Resale of ST products with provisions different from the information set forth herein shall void any warranty granted by ST for such product.

ST and the ST logo are trademarks of ST. All other product or service names are the property of their respective owners.

Information in this document supersedes and replaces information previously supplied in any prior versions of this document.

© 2017 STMicroelectronics - All rights reserved



DocID030724 Rev 1