STM32U585xx Security Target for PSA Certified™ Level 3 with SESIP Profile

Document information

This Security Target document is based on GlobalPlatform® Security Evaluation Standard for IoT Platforms [SESIP] methodology, version "Public Release v1.0". Document number: JSADEN011.

This document describes the STM32U585xx microcontrollers Security Target for PSA Certified™ Level 3 with SESIP Profile, included in the STM32CubeU5 v1.3.0 MCU Package.

1 About this document

1.1 Release information

Refer to Revision history.

1.2 References

This document refers to the following documents.

1.2.1 Normative references

Table 1. Normative references

Reference	Document Number	Author	Title
[PSA-L1]	JSADEN001	JSA	PSA Certified™ Level 1 Questionnaire
[PSA-EM-L2]	JSADEN003	JSA	PSA Certified™: Evaluation Methodology for PSA L2
[PSA-EM-L3]	JSADEN010	JSA	PSA Certified™: Evaluation Methodology for PSA L3
[PSA-AM]	JSADEN004	JSA	PSA Certified™ Attack Methods
[PSA-PP-L2]	JSADEN002	JSA	PSA Certified™ Level 2 Lightweight Protection Profile
[PSA-PP-L3]	JSADEN009	JSA	PSA Certified™ Level 3 Lightweight Protection Profile
[SESIP-PP-L2]	JSADEN012	JSA	SESIP Profile for PSA Certified™ Level 2
[PSA-L2-COMP]	JSADEN017	JSA	SESIP Profile for PSA Certified™ RoT Component Level 2
[PSA-L3-COMP]	JSADEN018	JSA	SESIP Profile for PSA Certified™ RoT Component Level 3
[SESIP]	GP_FST_070	GlobalPlatform®	Security Evaluation Standard for IoT Platforms (SESIP) v1.1
[CEM]	CCMB-2017-04-004	Common Criteria	Common Methodology for Information Technology Security Evaluation, Evaluation Methodology. Version 3.1, revision 5, April 2017.

1.2.2 Informative references

Table 2. Informative references

Reference	Document Number	Author	Title
[GP-ROT]	GP_REQ_025	GlobalPlatform®	Root of Trust definitions and requirements, version 1.1, public release, June 2018
[PSA-SM]	ARM DEN 0079	Arm®(1)	Arm® Platform Security Model

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

arm

1.2.3 Terms and abbreviations

This document uses the following terms and abbreviations (refer to [PSA-SM] and [PSA-L1]).

Table 3. Terms and abbreviations

Term	Meaning
Term	**Meaning**
Application	Used in this SESIP profile to refer to the components which are out of the scope of the evaluation.
Application Root of Trust Service(s)	Application-specific security service(s) that are not defined by PSA. Such services execute in the Secure Processing Environment and are required to be in Secure Partitions.
Application Specific Software	Software that provides the functionality required of the specific device. This software runs in the Non-Secure Processing Environment, making use of the System Software, Application RoT Services, and PSA-RoT Services.
Critical Security Parameter	Secret information, with integrity and confidentiality requirements, used to maintain device security, such as authentication data (passwords, PIN, certificates), and secret cryptographic keys.
Evaluation Laboratory	Laboratory or facility that performs the technical review of questionnaires submitted for Level 1 PSA certification. The list of evaluation laboratories participating in PSA Certified™ can be found on www.psacertified.org
Hardware Unique Key (HUK)	Secret and unique to the device symmetric key that must not be accessible outside the PSA Root of Trust. It is a Critical Security Parameter.
Host Platform	The entity which when used in composition with a certified PSA Level 2 RoT Component [PSA-L2-COMP] or a certified PSA Level 3 RoT Component [PSA-L3-COMP] form the scope of the certification covered in this profile.
Initial Attestation Key (IAK)	A PSA-RoT secret private key from an asymmetric key-pair used to sign attestation reports, thus ensuring that the report is bound to a unique PSA-RoT (and so device) instance.
Non-secure Processing Environment (NSPE)	The processing environment that hosts the non-secure System Software and Application-Specific Software. PSA requires the NSPE to be isolated from the SPE. Isolation between partitions within the NSPE is not required by PSA though is encouraged where supported.
Partition	The logical boundary of a software entity with intended interaction only via defined interfaces, but not necessarily isolated from software in other partitions. Note that both the NSPE and SPE may host partitions.
Platform	Used in this SESIP Profile to refer to the components which are in the scope of the evaluation.
PSA	Platform Security Architecture Document Number: JSADEN011 Version: 1.0 REL 02 Non-Confidential 9
PSA Certification Body	The entity that receives applications for PSA security certification, issues the certificates, maintains the security certification scheme, and ensures consistency across all the evaluation laboratories.
PSA Functional APIs	PSA defined Application Programming Interfaces on which security services can be built. APIs defined so far include Crypto, Secure Storage, and Attestation.
PSA Functional API Certification	Functional certification confirms that the device implements the PSA Functional APIs correctly by passing the PSA Functional certification test suites.
PSA Root of Trust (PSA-RoT)	PSA has defined a combination of the Immutable Platform Root of Trust and the Updateable Platform Root of Trust considered as being the most trusted security component on the device. Refer to [PSA-SM].
Immutable Platform Root of Trust	The minimal set of hardware, firmware, and data of the PSA-RoT, which is inherently trusted because it cannot be modified following manufacture. There is no software at a deeper level that can verify that it is as authentic and unmodified.

Term	Meaning
Updateable Platform Root of Trust	The firmware, software, and data of the PSA-RoT that can be securely updated following manufacture.
Platform Root of Trust Service(s)	PSA-defined security services for use by PSA-RoT, Application RoT Service(s), and the NSPE. Executes in the Secure Processing Environment and may use Trusted Subsystems. This includes the services offered by the PSA Functional APIs.
SESIP Profile	Document providing a common set of functionality for similar products
Secure Partition	A partition in the Secure Processing Environment.
Secure Processing Environment partition management	Management of the execution of software in Secure Partitions. Typical implementations will provide scheduling and inter-partition communication mechanisms. Implementations may also enforce isolation between the managed Secure Partitions.
Secure Processing Environment (SPE)	The processing environment that hosts the PSA-RoT, and any Application RoT Service(s). In SESIP terms, the SPE is the 'platform'.
Secure Boot	The process of verifying and validating the integrity and authenticity of updateable firmware and software components is a pre-requisite to their execution. This must also apply to all the firmware and software in the SPE. It must also apply to the first NSPE image loaded, which may extend the NSPE secure boot chain further.
Security Target (ST)	Document providing an implementation-dependent statement of security of a specific identified platform.
System Software	NSPE software that may comprise an Operating System or some run-time executive code, together with any middleware, standard stacks and libraries, chip-specific device drivers, etc., but not the application-specific software.
TOE	Target of Evaluation. In this SESIP Profile, it is a synonym for Platform.
Trusted subsystem	A security subsystem that the PSA-RoT relies on for the protection of its assets, or that implements some of its services.

1.3 PSA Certified™ Level 3

PSA defines a common hardware and software security platform, providing a generic security foundation and allowing secure products and features to be developed on top of this platform.

The PSA Certified™ scheme involves the evaluation by a laboratory of a device against a set of security requirements and, in case of a successful evaluation, the certification by the PSA Certified™ certification body of this Platform. The evaluation laboratory examines measures and processes to ensure that a functional Platform is not vulnerable to the identified threats to the levels defined in this document.

The PSA program recognizes that there will be different security requirements and different cost/security trade-offs for different applications and ecosystems. This is reflected in specifications by introducing a range of assurance levels.

Two evaluation paths are currently possible for a PSA Certified™ Level 3 product, either through the PSA Certified™ Level 3 Protection Profile [PSA-PP-L3] and associated evaluation methodology [PSA-EM-L3] [PSA-EM-L3], or through a SESIP evaluation using this SESIP Profile defined in this document.

1.3.1 PSA Certified™ RoT Component

The PSA Certified™ scheme allows for the certification of RoT Components that address a subset of the security functions required by an implementation for a Level 3 certifiable PSA Root of Trust (RoT) in accordance with this protection profile.

In the PSA Security Model [PSA-SM] such parts of a Root of Trust are referred to as a Trusted Subsystem. A typical example is an IP block that will be used in a chip. The IP could address a few security functions, with the rest of the chip covering all other requirements. Another example is an external chip that addresses a subset of the security functions, which when connected to another chip (Host Platform) form a complete Level 3 certifiable PSA-RoT.

A PSA L3 RoT Component [PSA-L3-COMP] may be used to aid the evaluation of an L3 PSA-RoT certification. A PSA L3 RoT Component may be used in composition for the evaluation of an L2 PSA-RoT certification and may be used to obtain a 'PSA Certified™ L2+SE' certificate (refer to [SESIP-PP-L2]).

2 Introduction

PSA Certified™ is the independent security evaluation scheme for Platform Security Architecture (PSA) based IoT systems. It establishes trust through a multi-level assurance program for chips containing a security component called a Root of Trust (PSA-RoT) that provides trusted functionality to the platform. The multi-level scheme has been designed to help device makers and businesses get the level of security they need for their use cases.

PSA Certified™ Level 3 is a fixed-time, test laboratory-based test, evaluation of the PSA-RoT. It is aimed at IoT devices that need to protect against substantial physical and software attacks. The Level 3 documents include: a SESIP Profile that describes the Target of Evaluation, its assets, the security objectives and security functions that will be evaluated, and an Attack Methods (AM) document describing the attacks in scope.

Developers submit their PSA-RoT to an approved test laboratory, listed on www.psacertified.org, for Level 3 evaluation and receive an Evaluation Technical Report. If the PSA-RoT is assessed as passing and approved by the independent Certification Body, a digital certificate will be issued on the PSA Certified™ website.

Keywords

PSA Certified™ Level 3, SESIP, Certification, IoT, Platform Security Architecture, Questionnaire, PSA, Security

The Security Target document describes the platform (in this chapter) and the exact security properties of the platform that are evaluated against SESIP assurance Level 3 (SESIP3) [SESIP] (in Section 4.2 Base PP security functional requirements) and that a potential consumer can rely upon the product upholding if they fulfill the objectives for the environment (in Section 3 Security objectives for the operational environment).

2.1 Security Target Reference

This document: Technical note *STM32U585xx Security Target for PSA Certified™ Level 3 with SESIP Profile* (TN1455), STMicroelectronics.

2.2 SESIP profile reference

Table 4. SESIP profile reference

Reference	Value
Protection profile name	SESIP profile for PSA Certified™ level 3
Protection profile version	V1.0 REL 02
Assurance claim	SESIP assurance level 3 (SESIP 3)
Optional and additional SFRs	• Secure encrypted storage (internal storage) • Field return of platform

2.3 Platform reference

The platform is uniquely identified by its chip (hardware) reference and its PSA-defined Root of Trust (software) reference as described below. The developer declares that only the evaluated and successfully certified products identify in this way.

Table 5. Platform reference

Reference	Value	
Platform name	STM32U585 TFM	
Platform version	1.3.0 based on TF-M open source version TF-M v1.3.0 and based on mcu_boot open source version v1.7.2	
Platform identification	Chip name and version	STM32U585 device family (die 482 revision W)
	PSA-RoT name and version	) based on TF-M open source version TF-Mv1.3.0 and based on mcu_boot open source version v1.7.2 SHA256 (en.stm32cubeu5-v1-3-0.zip)= 376d2c91b895259d5667e03d6384d30d54411216992c927a8dbfc70891f12926 SHA256 (TFM_SBSFU_Boot code binary (personalized data excluded)) = EB2775D1640B17610C5AA4FAD40EB11F2ACA9BD94042D3630BC899E6684229A3 SHA256 (updatable part of the secure code binary) = 0BDE2991C6FD792BBE3D66D567CEAFFC153159D1380E5E9033810E62BB8E7AF4
Platform type	Microcontroller platform with a TF-M compliant firmware for IoT applications	

2.4 Included guidance documents

The following documents are included with the platform:

Table 6. Guidance documents

Reference	Name	Version
[UM2852]	User manual *STM32U585xx security guidance for PSA Certified™ Level 3 with SESIP Profile*	Rev 3
[FW]	STM32CubeU5 MCU Package	V1.3.0
[TFM]	Open source TF-M UserGuide for v1.3.0: https://tf-m-user-guide.trustedfirmware.org/docs/releases/1.3.0.html	1.3.0
[MCU_BOOT]	Open source mcu_boot user guide information available at **https://mcuboot.com/**	1.7.2
[UM2851]	User manual *Getting started with STM32CubeU5 TFM application*	Rev 3
[RM0456]	Reference manual *STM32U5 series Arm®-based 32-bit MCUs*	Rev 4
[PSA_ST_API]	PSA storage API	V1.0.0
[PSA_CRYPTO_API]	PSA cryptography API	V1.0.0
[PSA_ATTESTATION_API]	PSA attestation API	V1.0.0
[PSA_FWU_API]	PSA firmware update API	V0.7
[AN4992]	Application note *STM32 MCUs secure firmware install (SFI) overview*	Rev 14
[UM2237]	User manual *STM32CubeProgrammer software description*	Rev 22

2.5 Platform functional overview and description

2.5.1 Platform type

Arm® Cortex®-M33 based microcontroller with integrated flash and SRAM memories and with a PSA-compliant firmware based on the open-source TF-M reference implementation.

2.5.2 Physical scope

The STM32U5 series microcontroller is a general-purpose MCU solution to provide a new optimal balance between performance, power, and security.

The IoT solution running on top of the microcontroller consists of a TF-M compliant with the PSA Certified™ Level 3 scheme that serves as a Root of Trust.

The TOE consists of a hardware microcontroller, a set of software files (comprising the source code), and guidance documents. The TOE hardware is shipped to the customer by STMicroelectronics. The TOE source code and guidance documents can be downloaded directly from the STMicroelectronics website. The format of the guidance documents is PDF.

2.5.3 Logical scope

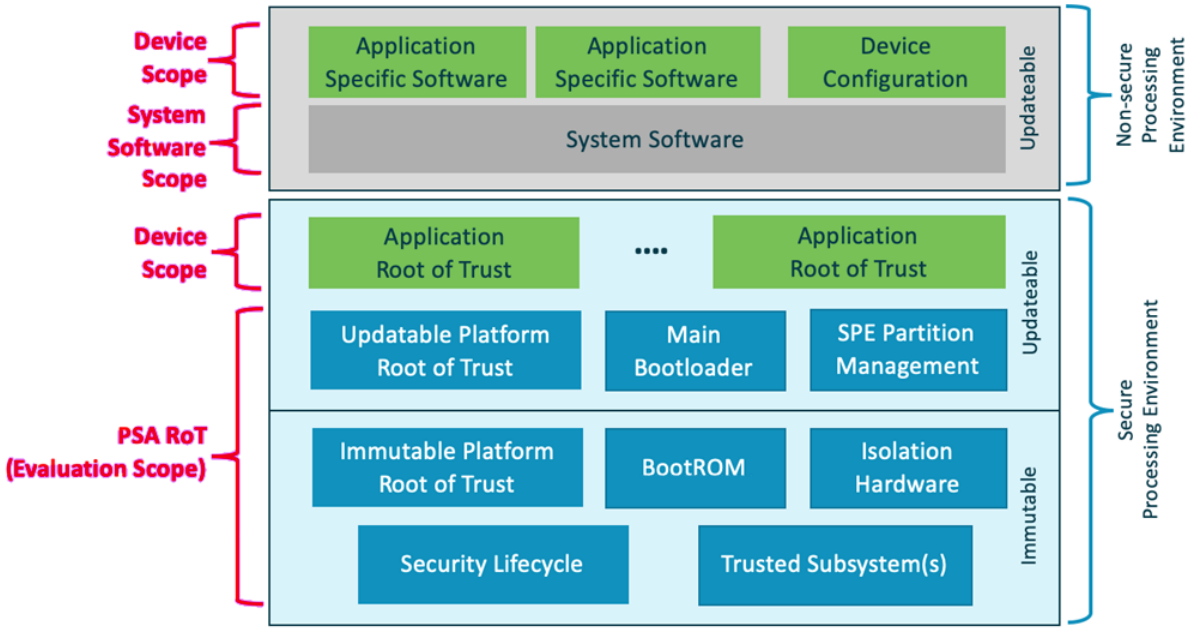
The scope for a PSA Certified™ Level 3 security evaluation, or Target Of Evaluation (TOE), is the combination of the trusted hardware and firmware components implementing a PSA-RoT with the security functional requirements stated in this document. PSA Certified™ Level 3 scope is identical to PSA Certified™ Level 2.

The chip security evaluation scope includes the following components as described in [PSA-SM]:

- Immutable platform Root of Trust, for example, the boot ROM, any root parameters, the isolation hardware, and hardware-based security lifecycle management and enforcement.
- Updateable platform Root of Trust, for example, can include the main bootloader code, the code that implements the SPE partition management function, and the code that implements the PSA defined services such as attestation, secure storage, and cryptography.

Trusted subsystems are components that the PSA Root of Trust relies on for the protection of its assets or implement some of its services, for example, a subscriber identification module or a secure element.

Figure 1. Scope of PSA Certified™ Level 3



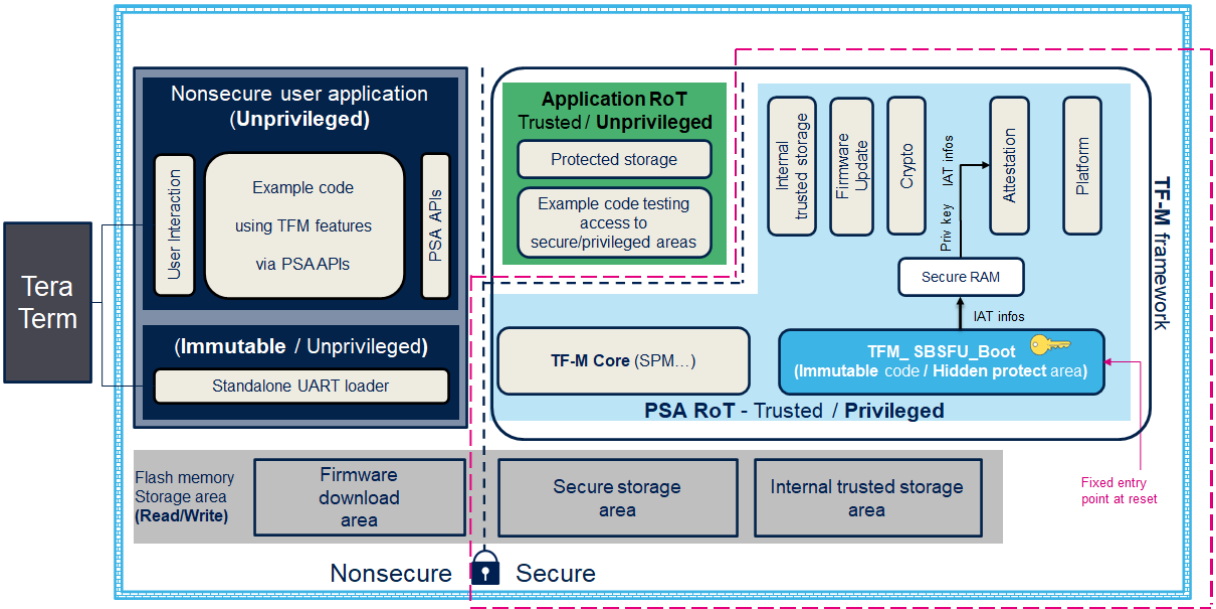
The TOE consists of a secure boot and secure firmware update application and consists of a set of secure services running on a STM32U5 series microcontroller, which are used to make a secure IoT product.

The secure boot and secure firmware update application and the set of secure services are implemented by porting the standard open-source mcu_boot firmware and the standard open-source TF-M trusted firmware on the STM32U5 microcontroller that brings the hardware security features needed to put in place the Root of Trust and to put in place the isolated domains.

The TOE is intended to be used by an integrator that deploys it into an IoT solution together with its own user application, providing assurance that the IoT application is securely booted, providing assurance that the IoT application can be securely updated, and providing the assurance that the secure part of the IoT application is well isolated from the non-secure part of the IoT application.

The physical scope is delimited by the fuchsia dotted line, as depicted in Figure 2, which comprises the TFM_SBSFU_Boot application, the TF-M core, the secure crypto services, the secure storage service (reduced to its file system only since the partition moved to the RoT application and is no more in the TOE scope), the Internal Trusted Storage service, the secure initial attestation service, and the secure image validation.

Figure 2. TOE scope



TF-M framework uses SMT32U5 hardware security features (especially CM33 TZ, MPU, and hide protect) to put in place 4 isolated domains:

- TFM_SBSFU_Boot application: secure privilege immutable code executed after reset managing the secure boot and the secure firmware update functions. This code is provisioned with some assets (RSA 2048 public Key for TF-M application authentication, RSA 2048 private key for decryption of the AES CTR symmetric key used to encrypt a new firmware image to be installed), it generates the IAT boot seed at each boot and it shares that information with TF-M trusted firmware via secure SRAM. This code and its associated assets are hidden just before the execution of the application using the temporal isolation mechanism. As a new feature, a few assets (e-g IAT private key) can be updatable and stored in secure SRAM when provisioned through a secure data image.
- Secure application
  - PSA-RoT: secure privilege code that can be updated by the TFM_SBSFU_Boot application. This code puts in place TF-M secure partitions and manages all the sensitive data and all the critical secure services. Secure services are exported to the non-secure application through the PSA APIs. Using v8-M TrustZone® and MPUs, TF-M controls the access to each TF-M secure partition by applications and other secure partitions and checks the validity of the parameters of any operation requested from applications.
  - Application RoT: secure non-privilege code that can be updated by the TFM_SBSFU_Boot application. This code is isolated from the PSA-RoT code as it is a non-privilege code and from the non-secure application but can be accessed by the non-secure application going through the PSA APIs that are controlled by TF-M.
- Non-Secure application: non-secure code that can only access secure services that are exported from the secure application through the PSA API. This code can be updated by TFM_SBSFU_Boot.

- Non-Secure standalone UART loader application: non-secure code that downloads in the non-secure firmware download area the new firmware images received through the UART interface. This code is immutable.

The source code of the TF-M framework is provided to the integrator. The integrator is also provided with development support tools to create user applications and with a user application example.

The integrator uses the security functionality provided by the TOE and can integrate its own secure non-privileged services in the RoT application to develop a secure IoT solution. The developer provides a demo application that serves as a test application for security functionalities.

2.5.4 Usage and major security features

The STM32U5 series targets the Internet of Things (IoT) medical, industrial, and consumer applications. It might be exposed to remote software attacks, or local attackers with limited resources, knowledge, or equipment. It might support connection to the network through a wired or wireless connection.

The PP considers the following features for PSA Level 3 security evaluation:

- A Secure Processing Environment (SPE) isolated by STM32U5 hardware mechanisms (TrustZone®) to protect critical services and related assets from the Non-Secure Processing Environment (NSPE).
- A secure boot process to verify the integrity and authenticity of executable code (but also provisioned data) and a secure firmware update application to be able to update the application and data in a secure way (authenticity check, integrity check, and version check). This code is the starting point of the Root of Trust as STM32U5 hardware mechanisms allow ensuring that this code is immutable (nonvolatile built-in flash memory protection (WRP) and Life cycle (RDP Level 2 with password capability allowing to go to RDPL1)) and allow ensuring that it is the first code executed after hardware reset (via boot-lock hardware mechanism). Related certificates are also protected in integrity as they cannot be modified thanks to the same STM32U5 hardware mechanisms.
- Support for secure storage, to protect the integrity and confidentiality of sensitive assets such as the RoT public key (ROTPK) and the attestation keys.
- Support for secure storage, to protect the integrity and confidentiality of sensitive assets for the SPE and related applications. Confidentiality is ensured by the encryption of sensitive data with the HUK (Hardware unique key).
- Support for Internal Trusted Storage, to write data in a STM32U5 built-in flash memory region that is isolated from the non-secure application or the non-secure/unprivileged application thanks to the STM32U5 security protection mechanisms.
- A security lifecycle for SPE, to protect the lifecycle state of the device and enforce the transition rules between states.
- Cryptographic functions services for SPE and NSPE applications.
- Support for entity attestation token (according to IETF specification).
- A temporal isolation feature that consists of a hardware mechanism offering an additional level of isolation for the immutable boot application and its associated assets.

2.5.5 Required hardware/software/firmware

No additional non-TOE hardware, software, or firmware is required.

3 Security objectives for the operational environment

To fulfill the platform's security requirements, the operational environment (technical or procedural) must fulfill the following objectives.

Table 7. Security objectives for the operational environment

ID	Description	Reference
TOE_SECRETS	The TOE secret keys used to protect the integrity and the authenticity of the installed user application or the user application firmware (code or data) updates need to be preserved under the custody of the user application developer.	Section 4.2.4 'Security Measures' of [UM2852].
TRUSTED_INTEGRATOR	The integrator builds/personalizes the TOE and uses the security functionalities needed by the user application following the TOE guidance documentation. The integrator is trusted and does not attempt to thwart the TOE security functionalities or bypass them.	Section 4.2.4 'Security Measures' of [UM2852].
TOE_PERSONALIZATION	The integrator provisions unique cryptographic keys and unique identifier inside TOE for each device using the TOE to ensure secure platform attestation and in to ensure secure encrypted storage.	Section 4.2.4 'Security Measures' of [UM2852].

4 Security requirements and implementation

4.1 Security assurance requirements

The claimed assurance requirements package is SESIP3, as defined in Section 5.1.

4.1.1 Flaw reporting procedure (ALC_FLR.2)

In accordance with the requirement for a flaw reporting procedure (ALC_FLR.2) including a process to give, generate any needed update, and distribute it, the developer has defined the procedure described in *https://www.st.com/content/st_com/en/security/report-vulnerabilities.html*.

4.2 Base PP security functional requirements

As a base, the platform fulfills the following security functional requirements:

4.2.1 Verification of platform identity

The platform provides a unique identification of the platform, including all its parts and their versions.

Conformance rationale:

The platform consists of hardware and firmware. The platform's unique identifier is provided in Section 2.3 and can be obtained via information that TOE provides through the PSA initial attestation services (*psa_initial_attest_get_token* function):

- Hardware version: contains the value of DBGMCU_IDCODE register that allows the identification of the STM32U5 hardware.
- Implementation ID: contains SHA256 value computed on the immutable SW code part of the TOE (TFM_SBSFU_Boot code binary data).
- Measurement value: contains SHA256 value computed on the updatable SW code part of the TOE (secure image code) and contains SHA256 value computed on the non-secure image code.

To uniquely identify the hardware, the platform uses the DBGMCU identity code register (DBGMCU_IDCODE) accessible to the debugger via the AHB access port:

- Hardware revision W (`0x3001`)
- STM32U585 device family (`0x482`)

4.2.2 Verification of platform instance identity

The platform provides a unique identification of that specific instantiation of the platform, including all its parts and their versions.

Conformance rationale:

In addition to the verification of platform identity, each TOE instance contains an immutable value that is unique per chip (SHA256 of a unique public key computed from a private key provisioned in the TOE as unique per chip). The unique identification of that specific instantiation of the platform can be obtained via information that TOE provides through the PSA initial attestation services (`psa_initial_attest_get_token` function): Instance ID = SHA256 of EAT public key, which is unique per TOE instance.

4.2.3 Attestation of platform genuineness

The platform provides attestation of the 'verification of platform identity' and 'verification of platform instance identity', in a way that ensures that the platform cannot be cloned or changed without detection.

Conformance rationale

The attestation is achieved through the token response, which is built with the challenge received by the application. The token is composed of the following elements:

- Boot-seed: random value generated at each boot by the `TFM_SBSFU_Boot` application
- Software measurement: HASH of firmware controlled and computed by the `TFM_SBSFU_Boot` application
- Implementation ID: digest (SHA256) of the immutable software code part of the TOE (`TFM_SBSFU_Boot` code binary data).
- Instance ID: digest (SHA256) of EAT public key (computed from a private key provisioned in the TOE as unique per chip).

- EAT public Key (computed from a private EAT key provisioned inside a secure data image)
- Hardware ID: immutable STM32U5 hardware version
- Life cycle: computed and verified by the `TFM_SBSFU_Boot` application and by the secure/privileged application.

The quantities provisioned inside the immutable part of the TOE, or computed by the secure boot function are issued to the secure/privileged application of the SPE:

- The challenge is communicated to the EAT service of the secure application (requestor)
- EAT function gets the information from the secure boot function and builds the token (with the challenge)
- Sign the token with the EAT private key (stored inside the secure SRAM)
- The non-secure application gets the token and can answer back to the original requestor

Hardware ID, implementation ID, software measurement, and instance ID information reported inside the signed token (computed from PSA initial attestation services) can be used by the verification entity to identify the platform type and the individual platform.

4.2.4 Secure initialization of platform

The platform ensures its authenticity and integrity during platform initialization. If the platform authenticity or integrity cannot be ensured, the platform will go to **a state where no other operation except optionally Secure Update of Platform can be performed**.

Conformance rationale

When the STM32U5 hardware lifecycle state is set to RDP level2 and the `Boot Lock` feature is set, the TOE boots on the immutable part of the internal flash, which hosts the `TFM_SBSFU_Boot` application code (secure boot) belonging to the TF-M firmware framework and the STM32U5 hardware ensures that data such as an instance ID located in the flash immutable area cannot be modified. Each time the system boots, the integrity and authenticity of SPE as well as NSPE are verified before execution.

The `TFM_SBSFU_Boot` application is started when the CPU is released from reset. It runs in Secure mode. It authenticates the firmware and data images by hash (SHA-256) and digital signatures (RSA-2048) validation.

The `TFM_SBSFU_Boot` application handles the secure and non-secure images independently (multiple image boot). The two application images (respectively the two data images) are signed independently with different keys and they can be updated separately.

Root parameters are either immutable (ROTPK, EAT public key, instance ID, and private key for decryption of firmware AES-CTR encryption key) as programmed in the immutable flash memory area or are computed (SHA256 of `TFM_SBSFU_Boot` application code, lifecycle state…) by the `TFM_SBSFU_Boot` application (trusted immutable code executed in secure mode) using immutable parameters and hardware immutable information, such as hardware version.

During the initialization process, in case the SPE or the NSPE is not verified OK (integrity not correct or authenticity not correct), then the `TFM_Boot` application executes an immutable non-secure loader application, which only allows downloading a new SPE and/or NSPE firmware version in the non-secure download slots.

During the initialization process, in case of any security configuration error, the system does not start to execute any application. Any violation results in a hardware reset or an infinite loop.

4.2.5 Attestation of platform state

The platform provides the attestation of the state of the platform, such that it can be determined that the platform is in a known state.

Conformance rationale

During the `TFM_SBSFU_Boot` application (that is part of the PSA immutable RoT) execution after each product reset, the product static security configuration is verified. In case of any security configuration error, the system does not start to execute any application. Once the SPE has been verified ok by the `TFM_SBSFU_Boot` application, the SPE execution is started and the SPE configures the dynamic security so that the initial attestation service of the SPE is executed from the secure/privileged domain. The initial attestation service reports the 'secure state' information inside the signed token as this service is only available once static security is correctly activated and once a verified SPE has been executed.

Any privilege violation during the execution of the `TFM_SBSFU_Boot` application or the execution of the SPE application is detected by the STM32U5 security hardware mechanisms and results in generating a hardware reset or in executing an infinite loop in the secure privileged domain.

4.2.6 Secure update of platform

The platform can be updated to a newer version in the field such that the integrity, authenticity, and confidentiality of the platform is maintained.

Conformance rationale

The `TFM_SBSFU_Boot` application (immutable part of the TOE) is started when the CPU is released from reset. It runs in Secure mode. It detects that there is a new SPE version installation request (this new SPE version has been upgraded in the SPE 'download' area located in the non-secure domain with one of these two installation methods: A new SPE version is either preloaded by the non-secure application or installed during the runtime as defined in the standard open-source TF-M trusted firmware, [PSA_FWU_API), decrypts the SPE image using AES-CTR cryptography based on a 128-bit symmetric key (retrieved from the new firmware image itself after decryption with an RSA private key), authenticates the SPE image by digital signature (RSA-2048) validation, checks the integrity by hash (SHA-256) and checks the SPE version to ensure an antirollback mechanism (rejects the installation of the old version). If all verifications are ok, then the `TFM_SBSFU_Boot` application installs the new SPE version (by copying the new SPE image from the SPE 'download' area to the SPE 'active' area, which is located in the secure domain). Once installed, the new SPE image is again verified at each boot before being executed by the `TFM_SBSFU_Boot` application.

`TFM_SBSFU_Boot` application handles the SPE and the NSPE images independently (multiple image boot). The two application images (respectively data images) are signed and encrypted independently with different keys and they can be updated separately.

Data confidentiality during the update operation is ensured as the SPE image is received encrypted (AES-CTR cryptography using a 128-bit symmetric key). However, confidentiality also relies on the following key points:

- Personalization may only be performed by a trusted party.
- Personalization of the product can be performed once the TOE is on the field. Therefore, personalization data (e.g. keys: IAT private key) must always be sent encrypted over an insecure network and with the conventional TLV format.

4.2.7 Physical attacker resistance

The platform detects or prevents attacks by an attacker with physical access before the attacker compromises any of the other functional requirements.

Conformance rationale

The TOE in the certified configuration only allows setting the RDP Level 2 with a password.

In RDP Level 2, debug connection is not possible and it is only possible to do an RDP regression to Level 1 when a password has been provisioned inside the STM32U5 hardware. In case there is no password programmed in the STM32U5 hardware then the product is locked in the RDP Level 2 configuration.

In RDP Level 1, it is not possible to access the TOE contents with JTAG debug interface or physical access, but it is possible to go to the product's virgin state (flash and protected memories are first erased before reopening the JTAG debug interface with full debug capabilities).

In both levels, the chip does not allow any physical modification of the RDP register.

TOE uses STM32U5 hardware peripherals allowing it to resist some physical attacks:

- DPA-resistant hardware crypto engines against side-channel attacks
- Antitamper hardware mechanisms for detecting voltage or frequency glitches

Moreover, the `TFM_SBSFU_Boot` application and SPE application embed software mitigations code (such as tests duplication or flow control) to protect a critical section of the secure code (such as dynamic security activation, image signature verification…) against perturbation attacks.

4.2.8 Software attacker resistance: Isolation of platform (between SPE and NSPE)

The platform provides isolation between the application and itself, such that an attacker able to run code as an application on the platform cannot compromise the other functional requirements.

Conformance rationale

The first level of isolation, NSPE versus SPE is guaranteed through the support of TZ in the TOE. This configuration is statically defined via STM32U5 option bytes. An additional hardware mechanism of temporal isolation (hide protect) allows adding another level of isolation inside the SPE that is used to hide the immutable part of the TOE (corresponding to the `TFM_SBSFU_Boot` application code and its associated assets) before executing the secure application.

4.2.9 Software attacker resistance: Isolation of platform (between PSA-RoT and application Root of Trust services)

The platform provides isolation between the application and itself, such that an attacker able to run code as an application on the platform cannot compromise the other functional requirements.

Conformance rationale

The second level of isolation, PSA-RoT versus Root of Trust application, is guaranteed through the support of isolation mechanisms in the TOE. A hardware MPU feature is configured in the secure domain to limit the access permissions, based on the separation of a privileged domain (for PSA_RoT) and an unprivileged domain (for application_RoT). Additional isolations are also provided by the PSA (TF-M) firmware framework (SPM part) to ensure that no data, or firmware that are part of the PSA_RoT can be modified in the application_RoT.

4.2.10 Cryptographic operation

The platform provides the application with Operations in Table 8 functionality with algorithms in Table 8 as specified in specifications in Table 8 for key lengths described in Table 8 and modes described in Table 8.

Rationale

Table 8. Cryptographic Operations

Algorithms	Cryptographic operations	Specification/Standard	Key lengths	Modes
AES	Encryption Decryption Authenticated encryption with associated data	NIST FIPS 197 NIST SP800-38A (ECB, CBC, CTR) NIST SP800-38D (GCM) IETF RFC 3610 (CCM)	128 or 256 bits	ECB (hardware) CBC (hardware) CTR (hardware) GCM (aead) (hardware) GMAC (aead) (hardware) CCM (aead) (hardware) CMAC (aead) (hardware) CFB (hardware)
RSA	Encryption Decryption Signature generation Signature verification	IETF RFC 8017 FIPS PUB 186-4	2048 or 3072 bits	Hardware implementation Encryption schemes: RSAES-OAEP, RSAES-PKCS1-v1_5, Signature scheme: RSASSA-PSS, RSASSA-PKCS1-v1_5, EMSA-PSS
ECC	ECDSA signature ECDSA verification	ANSI X9.62-2005	192, 224, 256, 384, 512, or 521 bits	EC curves (hardware): secp192r1, secp224r1, secp256r1, secp384r1, secp521r1, secp192k1, secp224k1, secp256k1, bp256r1, bp384r1, bp512r1 EC curves (software): curve25519, curve448
HASH	Secure hash(1) Keyed-hashing for message authentication (HMAC)	FIPS PUB180-4	Short or long key (HMAC only)	SHA2-224 (hardware), SHA2-256 (hardware), SHA2-384(software), SHA2-512(software) HMAC-SHA2-224 (hardware), HMAC-SHA2-256 (hardware), HMAC-SHA2-384(software), HMAC-SHA2-512(software)(2)

1. *For security reasons, the SHA1 algorithms must only be used for checksums and data integrity.*
2. *SHA2-512 includes reduced versions (SHA2-512/224 and SHA2-512/256.*

4.2.11 Cryptographic random number generation

The platform provides the application with a way based on a live entropy source (analog) to generate random numbers as specified in NIST SP 800-90B.

Conformance rationale

This security function is natively implemented inside the PSA firmware. TOE uses STM32U5xx hardware RNG IP (compliant with specification NIST SP800-90B) to provide a secure API to the NSPE to generate random numbers.

4.2.12 Cryptographic key generation

The platform provides the application with a way to generate cryptographic keys for use in cryptographic operations in Table 9 as specified in specifications in Table 9 for key lengths described in Table 9.

Rationale

Table 9. Cryptographic key generation

Algorithms	Cryptographic Key generation	Specification/Standard	Key lengths	Modes
DH	DH	RFC2631	RSA key 2048 and 3072 bits	RSA key pair generation (software)
ECDH	ECDH	ANSI X9.42	192, 224, 256, 384, 512, or 521 bits	EC curves (hardware): secp192r1, secp224r1, secp256r1, secp384r1, secp521r1, secp192k1, secp224k1, secp256k1, bp256r1, bp384r1, bp512r1 EC curves (software): curve25519, curve448

4.2.13 Cryptographic keystore

The platform provides the application with a way to store cryptographic keys defined in Table 10 such that not even the application can compromise the confidentiality of this data. This data can be used for the cryptographic operations defined in Section 4.2.10 Cryptographic operation.

Table 10. Cryptographic keystore

Iteration label	Cryptographic keys	Operation
IAT_prv	IAT private key	The token from the initial attestation secure service is signed with the IAT private key.
HUK	Hardware unique key	Used to encrypt data managed by TF-M secure storage service.
Non-secure application keys	Volatile or persistent cryptographic keys	The non-secure application can dynamically create volatile or persistent cryptographic keys inside the SPE, which can be later securely used via TF-M secure storage services.

Conformance rationale

A 256-bit HUK key is available inside the STM32U5 hardware and is directly connected with an internal bus to the STM32U5 secure AES hardware peripheral (thus cannot be read by any other peripheral such as the Arm® CPU core or DMA). The HUK is used when using the STM32U5 secure AES hardware peripheral in the context of the security function 'secure encrypted storage (internal storage)'.

IAT private key is programmed during the product manufacturing stagein the secure data primary slot region (that is in the secure/privileged domain and checked by the `TFM_SBSFU_Boot` application thanks to the `mcuboot` feature before executing the verified secure application).

IAT private key can also be downloaded, in an encrypted format in the non-secure data secondary slot region that is in a non-secure/privileged domain and that is processed (authenticity and integrity verification, anti-rollback check, decryption) before being copied by the `TFM_SBSFU_Boot` application in the secure data primary slot region.

Before executing the SPE, the `TFM_SBSFU_Boot` application verifies the IAT private key (TLV consistency). Only the SPE secure privilege part (containing the secure storage service and the initial attestation service) has the required privilege to access those secure areas.

Regarding the non-secure application cryptographic keys, the SPE provides different PSA APIs to create volatile or persistent cryptographic keys (for example by calling the `psa_import_key` service or by calling the `psa_generate_key` service) inside the secure privilege domain. The created key values are stored in the secure privilege memory and can be later securely used by the non-secure application through the TF-M crypto services. Once created, the key value is never disclosed by the SPE to the non-secure application and can only be used by referencing it with the identifier returned during the key creation service. When keys are stored in secure persistent memory, the keys are always available until they are destroyed.

4.3 Additional security functional requirements

Complete this section with the additional SFRs defined in [SESIP].

4.3.1 Field return of platform

The platform can be returned to the vendor without user data.

Conformance rationale

In the context of SESIP Level 3 certification, the final product configuration must use RDP Level 2 with a password. In RDP Level 2, STM32U5 hardware is still able to move to RDPL1 because a password has been provisioned inside the STM32U5 hardware.

When RDP is set to Level 1, flash and protected memories cannot be accessed via the JTAG interface, but it is still possible to do an RDP regression to level 0 to go to the product's virgin state (flash and protected memories are first erased before reopening the JTAG interface with full debug capabilities).

On the other hand, when RDP is set to Level 2 without a password programmed in the STM32U5 hardware, the product is locked and it is not possible to change the RDP level.

4.4 Optional security functional requirements

4.4.1 Secure encrypted storage (internal storage)

The platform ensures that all data stored by the application, except for data stored in the non-secure domain and data stored using the Internal Trusted Storage service (ITS), is encrypted as specified in the AES-GCM-based AEAD encryption policy with a platform instance unique 256-bit key.

Conformance rationale

Those security functions are natively implemented (secure storage service and Internal Trusted Storage service) inside the PSA firmware framework (ST-TF-M).

The sensitive assets are protected through the privileged code inside the SPE corresponding to the secure services implemented in TF-M framework:

- TF-M Secure Storage (SST) Service (implemented in the SPE/PSA_RoT part): provides confidentiality and integrity of assets. SST has a nonhierarchical storage model, such as a file system, where all the assets are managed by a linearly indexed list of metadata. The SST service implements an AES-GCM-based AEAD encryption policy to protect data integrity and authenticity. Secure storage flash areas are encrypted with the HUK (256 bits).
- TF-M Internal Trusted Storage (ITS) service (implemented in the SPE/PSA_RoT part): provides integrity and isolation (cannot be accessed directly from non-secure domain or secure unprivileged domain) of assets. The service has a nonhierarchical storage model, such as a file system, where all the assets are managed by a linearly indexed list of metadata. Contrary to the SST service, the ITS service does not implement any encryption policy, and the confidentiality of data is ensured thanks to hardware isolation of the internal flash access domain (secure/privileged).

Non-secure applications and secure non-privileged SPE codes cannot access directly these assets.

5 Mapping and sufficiency rationales

5.1 Assurance

The assurance activities defined in [PSA-EM-L3] fulfill the SESIP3 activities. In particular, the required source code review, vulnerability analysis, and testing to an equivalent of 35 person-days of the [PSA-EM-L3] is applicable.

Table 11. Assurance mapping and sufficiency rationales

Assurance class	Assurance Family	Covered by
ASE: Security Target evaluation	ASE_INT.1 Security Target Introduction	Section 2
	Rationale: The Security Target reference is in the title, the TOE reference in the 'platform reference', and the TOE overview and description in 'platform functional overview and description'.	
	ASE_OBJ.1 security requirements for the operational environment	Section 3
	Rationale: The objectives for the operational environment in 'security objectives for the operational environment'. Refer to the guidance documents.	
	ASE_REQ.3 listed security requirements	From Section 4.2 to Section 4.3
	Rationale: All SFRs in this Security Target are taken from [SESIP].	
	ASE_TSS.1 TOE summary specification	Section 4
	Rationale: All SFRs are listed per definition and for each SFR, the implementation and verification are defined in 'security functional requirements'.	
ADV: Development	ADV_FSP.4 complete functional specification	Section 2.4 and the material provided to the evaluator
	Rationale: The platform evaluator determines whether the provided evidence is suitable to meet the requirement.	
	ADV_IMP.3 complete mapping of the implementation representation of the TSF to the SFRs	The material provided to the evaluator
	Rationale: The platform evaluator determines whether the provided evidence is suitable to meet the requirement.	
AGD: Guidance documents	AGD_OPE.1 operational user guidance	Section 2.4
	Rationale: The platform evaluator determines whether the provided evidence is suitable to meet the requirement.	
	AGD_PRE.1 preparative procedures	Section 2.4
	Rationale: The platform evaluator determines whether the provided evidence is suitable to meet the requirement.	
ALC: Life-cycle support	ALC_CMC.1 labeling of the TOE	Section 2.4
	Rationale: The platform evaluator determines whether the provided evidence is suitable to meet the requirement.	
	ALC_CMS.1 TOE CM coverage	Section 1.2
	Rationale: The platform evaluator determines whether the provided evidence is suitable to meet the requirement.	
	ALC_FLR.2 flaw reporting procedures	Section 4.1.1
	Rationale: The flaw reporting and the remediation procedure are described.	
ATE: Tests	ATE_IND.1 independent testing: conformance	The material provided to the evaluator
	Rationale: The platform evaluator determines whether the provided evidence is suitable to meet the requirement.	

Assurance class	Assurance Family	Covered by
AVA: Vulnerability assessment	AVA_VAN.3 focused vulnerability analysis	NA A vulnerability analysis is performed by the platform evaluator to ascertain the presence of potential vulnerabilities.
	Rationale: The platform evaluator performs penetration testing, to confirm that the potential vulnerabilities cannot be exploited in the operational environment for the TOE. Penetration testing is performed by the platform evaluator assuming an attack potential of enhanced-basic.	

5.2 Functionality

Table 12. Functionality mapping and sufficiency rationales

PSA security function	Covered by SESIP SFR	Rationale
F.INITIALIZATION	Secure initialization of platform	Full coverage
F.SOFTWARE_ ISOLATION	Software attacker resistance: Isolation of platform between SPE and NSPE	Full coverage
	Software attacker resistance: Isolation of platform between PSA-RoT and Application Root of Trust Services	Full coverage
	Software attacker resistance: Isolation of application parts (between each of the Application Root of Trust Services)	Full coverage
F.SECURE_ STORAGE	Secure encrypted storage (internal storage)	Requires encryption mechanism providing both integrity and confidentiality.
	Secure storage (internal storage)	Requires authenticity and integrity
	Software attacker resistance: Isolation of platform (between SPE and NSPE)	Stored data is isolated from the NSPE and application Root of Trust Services by using a unique HUK for each platform.
	Not covered by any SESIP SFR. Note added in Secure encrypted storage (internal storage).	Covered by user guidance.
	Secure external storage	Requires encryption mechanism providing authenticity, integrity, and confidentiality.
F.FIRMWARE_ UPDATE	Secure update of platform	Full coverage
F.SECURE_STATE	Software attacker resistance: Isolation of platform (between SPE and NSPE)	Full coverage
	Software attacker resistance: Isolation of platform (between PSA-RoT and Application Root of Trust Services)	Full coverage
	Partially covered by the SFR 'secure initialization of the platform' and 'secure update of the platform'	Full coverage

PSA security function	Covered by SESIP SFR	Rationale
F.CRYPTO	Cryptographic operation	Additional algorithms can be added based on the supported algorithms provided by the PSA cryptographic API.
	Cryptographic keystores	Additional algorithms can be added based on the supported algorithms provided by the PSA cryptographic API.
	Cryptographic random number	The evaluation of the random number generator must follow a recognized methodology, e.g. AIS31 or SP800-90.
	Cryptographic key generation	Additional algorithms can be added based on the supported algorithms provided by the PSA cryptographic API.
F.ATTESTATION	Verification of platform identity	Unique identification of the platform
	Verification of platform instance identity	Unique identification of the platform instance
	Attestation of platform genuineness	'verification of platform instance' and 'verification of platform instance identity' are included in the attestation token.
	Attestation of platform State	Full coverage
F.AUDIT	Audit log generation and storage	NA The PSA implementation, including the chip, does not implement the generation of audit records. The STM32U5 does not implement this security function because of the flash size constraint. The objective is to optimize the SPE application to make sure enough memory remains available for the NSPE application.
F.DEBUG	Secure debugging	NA Not claimed in PSA because this feature is not accessible to the user.
F.PHYSICAL	Physical attacker resistance	Full coverage

Revision history

Table 13. **Document revision history**

Date	Revision	Changes
30-Jun-2021	1	Initial release. Final version for SESIP 3 certification.
27-Jun-2023	2	Moved to the 1.0 REL02 template Updated: • After Brightsight's review (U5 action item list v8.0) • Section 4.1.1 Flaw reporting procedure (ALC_FLR.2) • For STM32CubeU5 v1.3.0 MCU Package

Contents

List of tables

List of figures