Security Target for STM32U585 family of device compliant with SESIP Profile for PSA Certified™ Level 3

Based on [SESIP] methodology, version “Public Release v1.0”

Document number:	-
Version:	V2.0
Author	ST Microelectronics
Authorized by:	
Date of Issue:	21/07/2021

Contents

1 Introduction

The Security Target 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 chapter "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 chapter "Security Objectives for the operational environment").

1.1 ST Reference

See title page.

1.2 SESIP Profile Reference

Reference	Value
PP Name	SESIP Profile for PSA Certified Level 3
PP Version	V1.0ALP01
Assurance Claim	SESIP Assurance Level 3 (SESIP 3)
Optional and additional SFRs	• Secure Encrypted Storage (internal storage) • Field return of platform

Table 1: SESIP Profile Reference

1.3 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.

Reference	Value	
TOE Name	STM32U585 TFM	
TOE Version	1.0.0 (based on TF-M Open Source version TF-M v1.0-RC2 and based on mcu_boot Open Source version (SHA1=a40b19976158b8d0d1016ba82dcd4f7c896efe37)	
TOE Identification	Chip name and version	STM32U585 family of device (Die 482 Revision X)
	PSA-RoT name and version	Based on TFM Open Source version TF-MvV1.0-RC2 and based on mcu_boot Open Source version (SHA1=a40b19976158b8d0d1016ba82dcd4f7c896efe37): SHA256 (STM32Cube_FW_U585_ Security_certification v1_0_0.exe)= f114374be96608eb2b259a5da03653f9d30b596eb91d4b8637f93219ae900afe SHA256 (TFM_SBSFU_Boot code binary (Personalized data excluded)) = e3aec2284bb93b08474ecc2a1d1a2059fe5fde4fbb803268f1bc13738bc50156

		SHA256 (updatable part of the secure code binary) = 6f2ffbde4e6264165423342e83fdf18f559162dbe563e9bd89a507663f279d6d
TOE type	Microcontroller platform with a TF-M compliant firmware for IoT applications	

Table 2: Platform Reference

1.4 Included Guidance Documents

The following documents are included with the platform:

Reference	Name	Version
[UM2852]	STM32CubeU5 TFM security guidance for SESIP profile for ARM PSA Level 3 chip	Rev 1
[FW]	STM32Cube_FW_U585_Security_certification v1_0_0	V1.0.0
[TFM_RC2_RM]	Open Source TF-M User Guide for v1.0-RC2: https://ci.trustedfirmware.org/job/tf-m-build-test-nightly/lastSuccessfulBuild/artifact/build-docs/tf-m_documents/install/doc/user_guide/html/index.html	1.0RC2
[MCU_BOOT]	Open Source mcu_boot User guide information available at **https://mcuboot.com/**	NA
[UM2851]	Getting started with STM32CubeU5 TFM application	Rev 1
[RM0456]	STM32U5 Reference Manual	Rev 1
[PSA_ST_API]	PSA Storage API-1.0.0	V1.0.0
[PSA_CRYPTO_API]	PSA Cryptography API-1.0.0	V1.0.0
[PSA_ATTESTATION_API]	PSA Attestation API-1.0.0	V1.0.0
[AN4992]	Overview secure firmware install (SFI)	Rev 10
[UM2237]	STM32CubeProgrammer software description user manual	Rev 15

Table 3: Guidance Documents

1.5 Platform Functional Overview and Description

1.5.1 TOE 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.

1.5.2 TOE Physical Scope

The STM32U5 microcontroller series is 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 ST Microelectronics. The TOE source code and guidance documents can be downloaded directly from ST Microelectronics web site. The format of the guidance documents is PDF.

1.5.3 TOE 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 protection of its assets, or that implement some of its services, for example, a Subscriber Identification Module or a Secure Element.

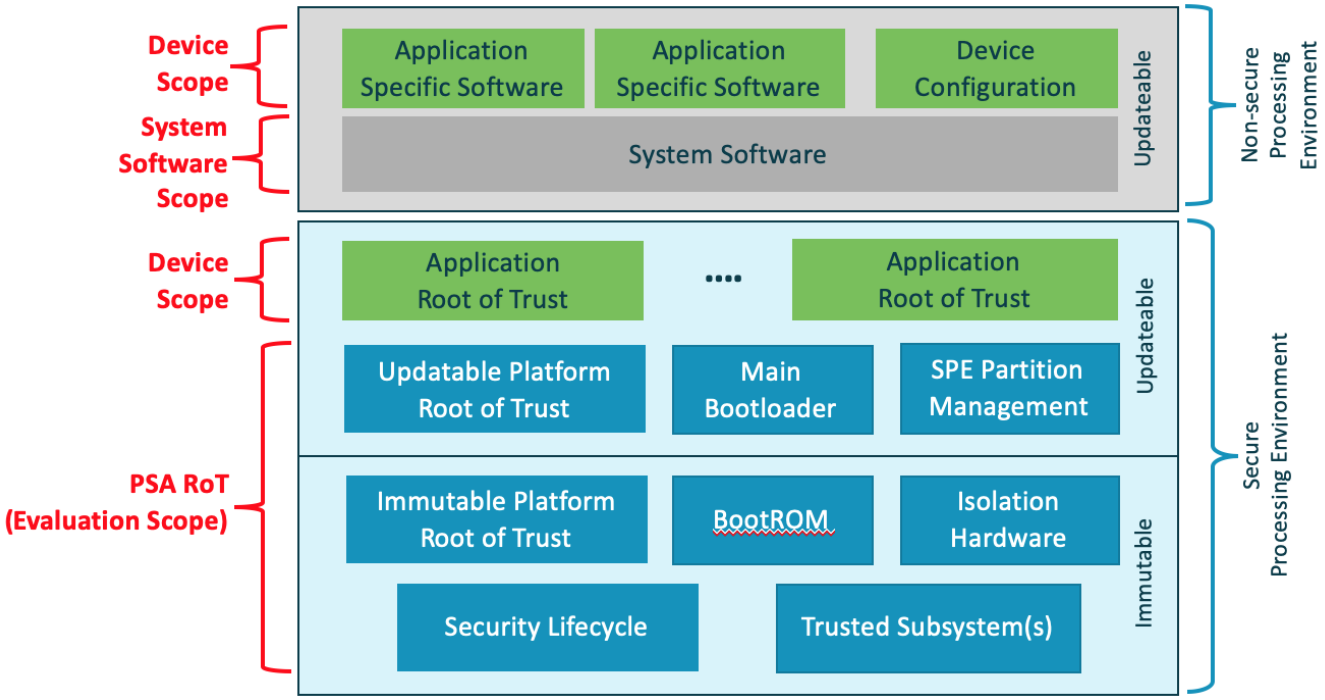

Figure 1: Scope of PSA Certified Level 3

The TOE consists of a secure boot and secure Firmware Update application and consists of a set of secure services running on a STM32U5 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 the 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 red dotted line, as depicted in Figure 2: TOE scope, which comprises the TFM_SBSFU_Boot application, the TF-M core, the secure crypto services, the secure storage service, the Internal Trusted Storage service, the secure initial attestation service and the Secure image validation.

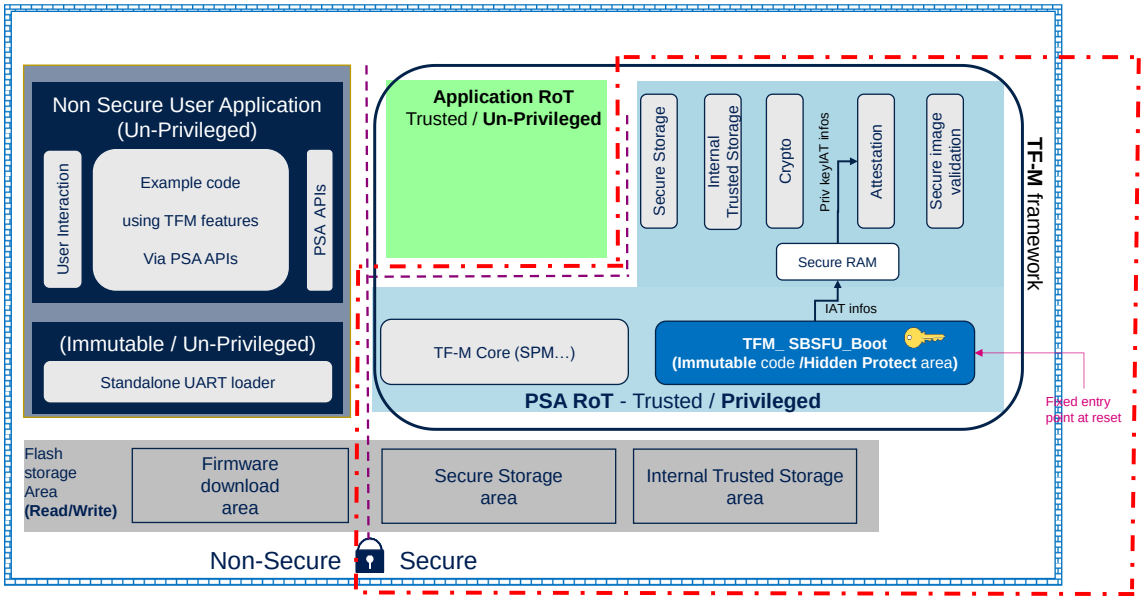

Figure 2: TOE scope

TFM 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 TFM application authentication, IAT private key, RSA 2048 private key for decryption of the AES CTR symmetric key used to encrypt a new firmware image to be installed), it will generate the IAT Boot seed at each boot and it will share those 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.
- Secure application
    - PSA RoT: secure privilege code that can be updated by 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 parameters of any operation requested from Applications.

  - Application RoT: secure non privilege code that can be updated by 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 will be 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 TFM framework is provided to the integrator. The integrator is also provided with development support tools to create user applications and with an user application example.

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

1.5.4 Usage and Major Security Features

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

The PP considers the following features for the purpose of 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 integrity and authenticity of executable code and a secure Firmware Update application in order to be able to update the application 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 to ensure that this code is immutable (non-volatile built-in flash memory protection (WRP) and Life cycle (RDP Level 2 with password capability allowing to go to RDPL1)) and allow to ensure that it is the first code executed after HW reset (via Boot lock HW mechanism). Related certificates are also protected in integrity as they can’t be modified thanks to the same STM32U5 hardware mechanisms.
- Support for Secure Storage, to protect in integrity and confidentiality sensitive assets such as the ROT Public Key (ROTPK) and the Attestation keys.
- Support for Secure Storage, to protect in integrity and confidentiality sensitive assets for the SPE and related applications. The confidentiality is ensured by 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 will be isolated from non-secure application or from non-secure/Unprivileged application thanks to the STM32U5 security protection mechanisms.
- A Security Lifecycle for SPE, to protect the lifecycle state for 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 which consist on an hardware mechanism offering an additional level of isolation for the immutable Boot application and its associated assets.

1.5.5 Non-TOE Hardware/Software/Firmware

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

2 Security Objectives for the operational environment

For the platform to fulfil its security requirements, the operational environment (technical or procedural) must fulfil the following objectives.

ID	Description	Reference
TOE_SECRETS	The TOE secret keys used to protect the integrity and the authenticity of the installed user application or of the the user application firmware updates need to be preserved under custody of the user application developer.	§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 nor bypass them.	§ 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 in order to ensure secure platform attestation and in order to ensure secure encrypted storage.	§ 4.2.4 "Security Measures" of [UM2852].

Table 4: Security Objectives for the Operational Environment

3 Security Requirements and Implementation

3.1 Security Assurance Requirements

The claimed assurance requirements package is **SESIP3** as described in Section 4.1.

3.1.1 Flaw Reporting Procedure (ALC_FLR.2)

In accordance with the requirement for a flaw reporting procedure (ALC_FLR.2), including a process to report flaw and generate any needed update and distribute it, the developer has defined the following procedure:

To report a security vulnerability impacting a STM32 product or solution, you should contact STM32-CyberTeam through stm32_psirt@st.com.

Your findings should include the following information:

- Full references of the product (full part number) or solution (software versions, tools versions, …)
- Detailed description of the vulnerability
- All instructions needed to reproduce the issue
- Impact of the reported vulnerability, including details of the exploit

Due to the sensitivity of vulnerability information, it is recommended to provide your findings through encrypted email using the below STM32-CyberTeam PGP/GPG Key.

Vulnerability management process

Security vulnerabilities related to STM products are managed by the STM32-CyberTeam through the following 4 stages process:

- Reporting: Any new issue being reported to the STM32-CyberTeam
- Evaluating: STM32-CyberTeam will evaluate the potential vulnerability, by confirming the issue, analyze it and set a priority for resolving it.
- Solving: STM32-CyberTeam will collaborate with division R&D to investigate a solution to mitigate or solve the issue. Depending on the component impacted (hardware, firmware, tools) the lead-time to bring that solution to the market may vary. At this stage, an internal ticket will be created and managed.
- Communicating: Once a solution is available (fix or mitigation), ST will manage communicating back. Depending on the nature of the vulnerability and the mitigation, an appropriate action will be communicated to ST customers. The disclosure may be public or could target a restricted list of customers.

Note that the TFM_SBSFU_Boot application cannot be updated. The reason is that the TOE is an efficient and small component and adding additional functionalities would increase the security threats and complexity.

3.2 Base PP Security Functional Requirements

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

3.2.1 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 a hardware and a firmware. The platform unique identifier, is provided in section 1.3 and can be obtained via information that TOE provides through the PSA Initial Attestation services (psa_initial_attest_get_token function):

- HW version: contains value of DBGMCU_IDCODE register that allow to identify the STM32U5 HW.
- 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 X (0x0x2001)
- STM32U585 family of device (0x482)

3.2.2 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” (see Section 3.2.1), each TOE instance contains an immutable value which 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.

3.2.3 Attestation of Platform Genuineness

The platform provides an attestation of the “Verification of Platform Identity” and “Verification of Platform Instance Identity”, in a way that cannot be cloned or changed without detection.

Conformance rationale

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

- Boot-seed: random value generated at each boot by TFM_SBSFU_Boot application
- SW measurement: HASH of firmware controlled and computed by the TFM_SBSFU_Boot application
- Implementation ID: digest (SHA256) of the immutable SW 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 the TFM_SBSFU_Boot immutable data region area)
- Hardware ID: immutable STM32U5 HW version

- Life cycle: computed and verified by the TFM_SBSFU_Boot application and by 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 get the information from the secure boot function, build the token (with the challenge)
- Sign the token with the EAT private key (stored inside the secure SRAM)
- The non-secure application get the token and can answer back to the original requestor

Hardware ID, implementation ID, SW 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.

3.2.4 Secure Initialization of Platform

The platform ensures its authenticity and integrity during the 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 (with Physical Attacker Resistance) can be performed**.

Conformance rationale

When the STM32U5 HW 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 TFM_SBSFU_Boot application code (secure boot) belonging to the TF-M firmware framework and STM32U5 HW ensures that data (such Instance ID…) located in the flash immutable area can’t be modified. Each time the system boots, integrity and authenticity of the SPE as well as NSPE is verified before execution.

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

TFM_SBSFU_Boot application handles the secure and non-secure images independently (multiple image boot). The 2 images are signed independently with different keys and they can be updated separately.

Root parameters are either immutable (ROTPK, EAT public key, Instance ID, private key for decryption of FW 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 TFM_SBSFU_Boot application (trusted immutable code executed in secure mode) using immutable parameters and HW immutable information (HW version…).

During all the initialization process, in case the SPE or the NSPE have not been verified OK (integrity not correct or authenticity not correct), then TFM_Boot application will execute an immutable non-secure loader application which will only allow to download a new SPE and/or NSPE firmware version in the non-secure download slots.

During all the initialization process, in case of any security configuration error, the system will not start to execute any application. Any violation will result in a HW reset or an infinite loop.

3.2.5 Attestation of Platform State

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

Conformance rationale

During 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 will not start to execute any application. Once the SPE has been verified ok by the TFM_SBSFU_Boot application, the SPE execution will be started and the SPE will configure the dynamic security so that the initial attestation service of the SPE will be executed from the secure/privilege domain. The initial attestation service will report the “secure state” information inside the signed Token as this services 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 during the execution of the SPE application will be detected by the STM32U5 security HW mechanisms and will result in generating a HW reset or in executing an infinite loop in secure privileged domain.

3.2.6 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

TFM_SBSFU_Boot application (immutable part of the TOE) is started when CPU is released from reset. It runs in secure mode. It detects that there is a new SPE version installation request (new SPE version has been pre-loaded by the non-secure application in the SPE “download” area located in the non-secure domain), decrypts the SPE image using AES-CTR cryptography based a 128 bit symmetric key (retrieved from new FW image itself after decryption with 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 anti-rollback mechanism (rejects installation of old version). if all verifications are ok then 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 will be 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 2 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 following key points:

- Personalization can only be done by a trusted integrator.
- Personalization of the product cannot be performed once the TOE is on the field. Therefore, personalization data (e.g. keys) are never sent over an insecure network.

3.2.7 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, ensuring that the other functional requirements are not compromised.

Conformance rationale

The TOE in the certified configuration only allows to set the RDP Level 2.

In RDP Level 2, debug connection is not possible and it is only possible to do a RDP regression to Level 1 if a password has been provisioned inside the STM32U5 HW. In case there is no password programmed in the STM32U5 HW then the Product is locked in 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 product virgin state (Flash and protected memories will be 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 HW peripherals allowing to resist against some physical attacks:

- DPA resistant HW crypto engines against side-channel attacks.
- Anti-tamper HW mechanisms for detecting voltage glitch or frequency glitch

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

3.2.8 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

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 to add 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 to its associated assets) before executing the secure application.

3.2.9 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 is provided by the PSA (TF-M) firmware framework (SPM part) which implemented version answers to the level2 software isolation requirements by configuring the HW MPU feature and by updating HW TrustZone configuration of STM32U5 in order to define a privilege domain (for PSA_RoT) and an unprivileged domain (for application_RoT) in the secure domain.

3.2.10 Cryptographic Operation

The platform provides the application with **Operations in** Table 5 functionality with **algorithms in** Table 5 as specified in **specifications in** Table 5 for key lengths **described in** Table 5 and modes **described in** Table 5 .

Rationale

Table 5

Algorithms	**Cryptographic operations**	**Specification / Standard**	**Key size(s)**	**Mode(s)**
AES	Encryption Decryption Authenticated encryption with associated data	NIST FIPS 197 NIST SP800-38A (ECB, CBC, CTR) NIST SP800-38D (GCM)	128, 256 bits	ECB (HW) CBC (HW), CTR (HW), GCM (aead) (HW), GMAC (aead) (HW) , CCM (aead) (HW), CMAC (aead) (HW), CFB (HW)

		IETF RFC 3610 (CCM)		
RSA	Encryption Decryption Signature generation Signature verification	IETF RFC 8017 FIPS PUB 186-4	2048, 3072	HW 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, 521 bits	EC Curves (HW): secp192r1, secp224r1, secp256r1, secp384r1, secp521r1, secp192k1, secp224k1, secp256k1, bp256r1, bp384r1, bp512r1 EC Curves (SW): curve25519, curve448
HASH	Secure hash (1) Keyed-hashing for message authentication (HMAC)	FIPS PUB180-4	Short or long key (HMAC only)	SHA2-224 (HW), SHA2-256 (HW), SHA2-384(SW), SHA2-512(SW) HMAC-SHA2-224 (HW), HMAC-SHA2-256 (HW), HMAC-SHA2-384(SW), HMAC-SHA2-512(SW) (2)

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

3.2.11 Cryptographic Random Number Generation

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

Conformance rationale

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

3.2.12 Cryptographic Key Generation

The platform provides the application with a way to generate cryptographic keys for use in **cryptographic operations in** Table 6 as specified in **specifications in** Table 6 for key lengths **described in**Table 6.

Rationale

Table 6

Algorithms	Cryptographic Key Generation	Specification / Standard	Key size(s)	Mode(s)
DH	DH	RFC2631	RSA key 2048 and 3072 bits	RSA key pair generation (SW)
	ECDH	ANSI X9.42	192, 224, 256, 384, 512, 521 bits	EC Curves (HW): secp192r1, secp224r1, secp256r1, secp384r1, secp521r1, secp192k1, secp224k1, secp256k1, bp256r1, bp384r1, bp512r1 EC Curves (SW): curve25519, curve448

3.2.13 Cryptographic KeyStore

The platform provides the application with a way to store **Cryptographic keys defined in Table 7** such that not even the application can compromise the **confidentiality** of this data. This data can be used for the cryptographic operations **defined in chapter "3.2.10 Cryptographic Operation "**Cryptographic Operation**.**

Table 7 Cryptographic keystore

Iteration label	Cryptographic keys	Operation(s)
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 cryptographic keys	Non-secure application can dynamically create volatile 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 HW and is directly connected with an internal bus to the STM32U5 secure AES HW peripheral (thus can't be read by any other peripheral such as AMR CPU core or DMA). The HUK is used when using the STM32U5 secure AES HW peripheral in the context of the security function "Secure Encrypted Storage (internal storage)".

IAT private key is programmed during product manufacturing stage in the immutable TFM_SBSFU_Boot data region (that is in secure/privileged domain and that will be hidden by TFM_SBSFU_Boot application thanks to the STM32U5 hardware hide Protect feature before executing the verified secure application).

Before executing the SPE, the TFM_SBSFU_Boot application shares the IAT private key with the SPE via secure/privilege areas (SRAM and Back-up registers). Only the SPE secure privilege part (containing the secure storage service and the initial attestation service) have the required privilege to access to those secure areas.

Regarding the non-secure application cryptographic keys, the SPE provides different PSA APIs to create volatile cryptographic keys (for example by calling psa_import_key service or by calling psa_generate_key service) inside the secure privilege domain. The created keys value are stored in the secure privilege volatile 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. As keys are stored in secure privilege volatile memory, the keys are lost as soon as a power lost or a reset occurs.

3.3 Optional Security Functional Requirements

3.3.1 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 **AES-GCM based AEAD encryption policy** with a platform instance unique key of key length **256 bits**.

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 privileged code inside the SPE corresponding to the Secure Services implemented in TF-M framework:

- TF-M's Secure Storage (SST) Service (implemented in SPE/PSA_RoT part): provides confidentiality and integrity of assets. SST has a non-hierarchical storage model, as a filesystem, where all the assets are managed by 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 area are encrypted with the HUK (256 bits).
- TF-M's Internal Trusted Storage (ITS) service (implemented in SPE/PSA_RoT part): provides integrity and isolation (can't be accessed directly from non-secure domain or from Secure Unprivileged domain) of assets. The service has a nonhierarchical storage model, as a filesystem, where all the assets are managed by a linearly indexed list of metadata. Contrary to SST service, the ITS service does not implement any encryption policy, the confidentiality of data being ensured thanks hardware isolation of the internal flash access domain (secure/privilege).

Non-secure application and secure non-privilege SPE code can't access directly these assets

3.4 Additional Security Functional Requirements

3.4.1 Field return of platform

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

In the context of SESIP L3 certification, the final product configuration shall use RDP Level 2. In RDP Level 2, STM32U5 HW is still able to move to RDPL1 in case a password has been provisioned inside the STM32U5 HW.

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

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

4 Mapping and Sufficiency Rationales

4.1 Assurance

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

Assurance Class	Assurance Family	Covered by	Rationale
ASE: Security Target evaluation	ASE_INT.1 ST Introduction	Section “Introduction” and title page of the Security Target	The ST reference is in the “ST Reference”, the TOE reference in the “Platform Reference” , the TOE overview and description in “Platform Functional Overview and Description”.
	ASE_OBJ.1 Security requirements for the operational environment	Section “Security Objectives for the Operational Environment” of the Security Target	The table listing the objectives for the operational environment refers to the guidance documents.
	ASE_REQ.3 Listed Security requirements	Section “Security Requirements and Implementation” of the Security Target	All SFRs in the profile are taken from [SESIP]. “Verification of Platform Identity” is included. “Secure Update of Platform” is included.
	ASE_TSS.1 TOE Summary Specification	Section “Security Requirements and Implementation” of the Security Target	All SFRs are listed per definition, and for each SFR the implementation and verification is defined in “Base PP Security Functional Requirements”, “Optional Security Functional Requirements” and “Additional Security Functional Requirements”.
ADV: Development	ADV_FSP.4 Complete functional specification	Functional specification as specified in “References”	The functional specification describes the complete set of TSF interfaces.
	ADV_IMP.3 Complete mapping of the implementation	Implementation representation and mapping	The implementation representation can be mapped to the SFRs

	representation of the TSF to the SFRs	to SFRs as specified in "References"	defined in section "Base PP Security Functional Requirements", "Optional Security Functional Requirements" and "Additional Security Functional Requirements".
AGD: Guidance documents	AGD_OPE.1 Operational user guidance	Guidance listed in section "Included Guidance Documents"	The operational user guidance describes secure usage of the user accessible functions.
	AGD_PRE.1 Preparative procedures	Guidance listed in section "Included Guidance Documents"	The preparative procedures describe how the TOE is brought into a secure configuration.
ALC: Life-cycle support	ALC_CMC.1 Labelling of the TOE	Section "Platform Reference"	The TOE is clearly identified as stated in the ST.
	ALC_CMS.1 TOE CM Coverage	ALC documentation as specified in "References"	Configuration items are properly identified.
	ALC_FLR.2 Flaw reporting procedures	Flaw remediation documentation as specified in "References"	The flaw reporting and remediation procedure is described.
ATE: Tests	ATE_IND.1 Independent testing: conformance	PSA Functional API test log as specified in "References" and additional evaluator testing.	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.
AVA: Vulnerability Assessment	AVA_VAN.3 Focused vulnerability analysis	Vulnerability and testing carried out by the laboratory	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.

Table 8: Assurance Mapping and Sufficiency Rationales

4.2 Functionality

Table 9 Functionality Mapping and Sufficiency Rationales

PSA Security Function	(detail)	Covered by SESIP SFR	Rationale
F.INITIALIZATION	Boot sequence in scope: Chip PSA Root of Trust Application Root of Trust Services	Secure initialization of platform	Full coverage
F.SOFTWARE_ ISOLATION	Level 1: Isolation between SPE and NSPE	Software Attacker Resistance: Isolation of Platform (Level 1)	Full coverage
	Level 2: Isolation between PSA-RoT and Application RoT	Software Attacker Resistance: Isolation of Platform (Level 2)	Full coverage
	(Optional) Level 3: Isolation between Application RoT	Software Attacker Resistance: Isolation of Application Parts	Full coverage
F.SECURE_ STORAGE	Integrity and Confidentiality (optional)	Secure Encrypted Storage (internal storage)	Requires encryption mechanism providing both integrity and confidentiality.
	Authenticity and integrity (optional)	Secure Storage (internal storage)	Requires authenticity and integrity
	Binding to the RoT	Software Attacker Resistance: Isolation of Platform	Stored data is isolated from the NSPE and Application Root of Trust Services by using a unique HUK for each platform.
	Basic rollback – atomicity	Not covered by any SESIP SFR. Note added in “Secure Encrypted Storage”.	Covered by user guidance.
	External storage (optional)	Secure External Storage	Requires encryption mechanism providing authenticity,

			integrity and confidentiality.
F.FIRMWARE_ UPDATE	Integrity and authenticity of the update.	Secure Update of Platform	Full coverage
F.SECURE_STATE	Protects itself against abnormal situations caused by programmer errors or violation of good practices from code executed outside of the TOE, either from SPE or NSPE.	Software Attacker Resistance: Isolation of Platform	Full coverage
	Controls the access to its services by Applications and checks the validity of parameters of any operation requested from Applications	Software Attacker Resistance: Isolation of Platform	Full coverage
	Enters a secure state upon platform initialization error or software failure detection, without exposure of any sensitive data.	Partially covered by the SFR “Secure initialization of platform”, “Secure update of platform” and also for the TF-M implementation covering the software failure detection.	Full coverage
F.CRYPTO	Minimum cryptographic operations supported: Attestation Secure Storage	Cryptographic Operation	Additional algorithms can be added based on the supported algorithms provided by the PSA cryptographic API.
	Minimum cryptographic keys for secure storage: Attestation Secure Storage	Cryptographic KeyStore	Additional algorithms can be added based on the supported algorithms provided by the PSA cryptographic API.

	PSA SM requires that all devices implement at least the following trusted cryptographic services: True random number generator Global nonce counter	Cryptographic Random Number	The evaluation of the random number generator shall 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
	Unique platform number	Verification of Platform Instance Identity	Unique identification of the platform instance
	Proof of origin	Attestation of Platform Genuineness	“Verification of Platform Instance” and “Verification of Platform Instance Identity” are included in the attestation token.
	Lifecycle state	Attestation of Platform State	Full coverage
F.AUDIT	(Optional) Protect the stored audit records from unauthorized deletion	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 flash size constraint. The objective is to

			optimize the SPE application in order to make sure enough memory will remain available for the NSPE application.
	(Optional) Prevent unauthorized modifications	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 flash size constraint. The objective is to optimize the SPE application in order to make sure enough memory will remain available for the NSPE application.
F.DEBUG	Optional	Secure Debugging	NA Not claimed in PSA because this feature is not accessible to the user.
		Physical attacker Resistance	Debug is locked, so this SFR is covered by the other SFRs and the VA.
F.PHYSICAL		Physical Attacker Resistance	Full coverage

5 About this document

5.1 Release Information

The change history table lists the changes that have been made to this document.

Date	Version	Author	Change
2021-06-30	1.0	ST Microelectronics	Final version for SESIP 3 certification
2021-07-21	2.0	ST Microelectronics	Table in Section 4.1 completed. Comments from the profile removed.

5.2 References

This document refers to the following documents.

5.2.1 Normative references

Ref	Doc No	Author(s)	Title
[PSA-EM-L2]	JSADEN003	JSA	PSA Certified: Evaluation Methodology for PSA L2 v1.1
[PSA-EM-L3]	JSADEN010	JSA	PSA Certified: Evaluation Methodology for PSA L3 v1.0-ALP01
[PSA-AM-L2]	JSADEN004	JSA	PSA Certified Attack Method for PSA L2 v1.1
[PSA-AM-L3]	JSADEN008	JSA	PSA Certified Attack Method for PSA L3 v1.0-ALP01
[PSA-PP-L2]	JSADEN002	JSA	PSA Certified Level 2 Lightweight Protection Profile v1.1
[PSA-PP-L3]	JSADEN009	JSA	PSA Certified Level 3 Lightweight Protection Profile v1.0-ALP01
[SESIP]	GP_FST_070	GlobalPlatform	Security Evaluation Standard for IoT Platforms (SESIP) v1.0
[CEM]	CCMB-2017-04-004		Common Methodology for Information Technology Security Evaluation, Evaluation Methodology. Version 3.1, revision 5, April 2017.

5.2.2 Informative references

Ref	Doc No	Author(s)	Title
[GP-ROT]	GP_REQ_025	GlobalPlatform	Root of Trust Definitions and Requirements, Version 1.1, Public Release, June 2018
[JIL-APSC]	-	JHAS	Joint Interpretation Library – Application of Attack Potential to Smartcards v3.0 April 2019

[PSA-SM]	ARM DEN 0079	ARM	Platform Security Architecture Security Model v1.0

5.2.3 Evaluation references

Ref	Doc No	Author(s)	Title
[ADV_FSP]	-	STMicroelectronics	U585 TFM ADV_FSP mapping v1.1
[ADV_IMP]	-	STMicroelectronics	U585 TFM ADV_IMP mapping v1.1
[ALC]	-	STMicroelectronics	U585_TFM ALC_CMC&CMS v2.0
[ATE]	-	STMicroelectronics	U585 API compliance test log V1.1

5.3 Terms and Abbreviations

This document uses the following terms and abbreviations (see PSA-SM and PSA Cert L1 V2.1 or newer questionnaire).

Term	Meaning
Application	Used in SESIP to refer to the components which are out of the scope of the evaluation. It is a synonym for Connected Application.
Application Root of Trust Service(s)	Application specific security service(s), and so not defined by PSA. Such services execute in the Secure Processing Environment 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.
Connected Application	Software developed by an IoT vendor, implementing IoT end-user use case based on the underlying Connected Platform. May be referred to as “Application” when there is no ambiguity.
Connected Platform	Combination of hardware and software that provides a runtime environment for a Connected Application. A Connected Platform implements security features and makes security services available to the Connected Application. May be referred to as “platform” when there is no ambiguity.
Connected product	Combination of a Connected Platform and a Connected Application that a product vendor puts on the market.May be referred as “product” when there is no ambiguity.

Critical Security Parameter	Secret information, with integrity and confidentiality requirements, used to maintain device security, such as authentication data (passwords, PIN, certificates), secret cryptographic keys, etc..
Evaluation Laboratory	Laboratory or facility that performs the technical review of questionnaires submitted for Level 1 PSA certification. The list of evaluation laboratories participating to 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.
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. In SESIP terms, the NSPE is the “application”.
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 SESIP to refer to the components which are in the scope of the evaluation. It is a synonym for Connected platform.
Product	Used by SESIP as a synonym for Connected product
PSA	Platform Security Architecture
PSA Certification Body	The entity that receives applications for PSA security certification, issues 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)	The PSA defined combination of the Immutable Platform RooT of Trust and the Updateable Platform Root of Trust, and considered to be the most trusted security component on the device. See [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 as authentic and unmodified.
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 by the NSPE. Executes in the Secure Processing Environment and may use Trusted Subsystems. This includes the services offered by the PSA Functional APIs.
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 as a pre-requisite to their execution. This must apply to all the firmware and software in the SPE. It should also apply to the first NSPE image loaded, which may extend the NSPE secure boot chain further.
System Software	NSPE software that may comprise an Operating System or some run-time executive, together with any middleware, standard stacks and libraries, chip specific device drivers, etc., but not the application specific software.
Trusted subsystem	A security subsystem that the PSA-RoT relies on for protection of its assets, or that implement some of its services.