Security Target for STM32L5xx compliant with SESIP profile for PSA Level 2 chip

Version 1.0 P10, dated 2020-08-04

ST Microelectronics

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

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 GP] (in chapter "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 SESIP Profile Reference

This Security Target claims conformance to the SESIP Profile for PSA L2 [SESIP PP]. This SESIP Profile describes the requirements sufficient to fulfil the hardware requirements for a chip of the PSA Level 2 requirements as described in [PSA L2 PP]. Note that [SESIP PP] use the SFRs catalogue from [SESIP GP].

1.2 ST Reference

See title page.

1.3 Platform Reference

The TOE is a microcontroller with a TF-M compliant firmware.

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

TOE name	STM32L5xx
TOE version	1.1.0 (based on TF-M Open Source version TF-M v1.0-RC2)
TOE identification	STM32L5xx Die 472 Revision B
	Based on TFM Open Source version TF-MvV1.0-RC2: SHA256 (en.stm32cubel5_v1-1-0.zip)= 85936d6cdc2b24fd10f869b0d31db4caac09ad094961b6a83297ed02e7aae110 SHA256 (TFM_SBSFU code binary data) = f67578893f905ec579bb6aa237d5ad7c033159a9aa9e5aeda237ff4261b56b8a SHA256 (updatable part of the secure image code) = 635dd95abb7349cf05d74a3faa4f771cdbc8766f0e10427040951bf673a9138e
TOE Type	Microcontroller platform with a TF-M compliant firmware for IoT applications

1.4 Included Guidance Documents

The following documents are included with the platform:

Reference	Name	Version
[UM2745]	STM32CubeL5 TFM security guidance for SESIP profile for ARM PSA Level 2 chip	Rev.1
[FW]	STM32Cube_FW_L5_V1.1.0	1.1.0
[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
[AN5394]	Getting started with projects based on the STM32L5 Series in STM32CubeIDE	V 2
[UM2671]	Getting started with STM32CubeL5 TFM application	Rev.1

[UM2237]	STM32CubeProgrammer software description user manual	Rev.11
[UM2656]	Getting started with STM32CubeL5 for STM32L5 Series	Rev.1
[AN4992]	Overview secure firmware install (SFI) Application Note	Rev.7
[RM0438]	STM32L5 Reference Manual	Rev 5
[PSA_ST_API]	PSA Storage API-1.0.0	1.0.0
[PSA_CRYPTO_API]	PSA Cryptography API-1.0.0	1.0.0
[PSA_ATTESTATION_API]	PSA Attestation API-1.0.0	1.0.0

1.5 Other Certification

The product has previously been evaluated following the [PSA L2 EM]:

Scheme	PSA Certified L2
Certification body	TrustCB
Certification number	071605359631-10010
Certification date	18/02/2020

1.6 Platform Functional Overview and Description

1.6.1 TOE type

Microcontroller with a TF-M compliant firmware.

1.6.2 TOE Physical Scope

The STM32L5 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 2 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.

1.6.3 TOE Logical Scope

The scope for a PSA Level 2 Security evaluation, or Target of Evaluation (TOE), is the combination of the hardware and firmware components supporting a device compliant with PSA specification. The considered hardware may be a System-in-Package (SiP), a System-on-Chip (SoC) integrated on a board, or similar set-up.

The hardware is in the scope of the security evaluation as it provides security features, such as immutable storage or protection of JTAG, which are essential for ensuring the security of the PSA implementation.

The PSA platform components that are in the scope of the security evaluation, as described in [PSA-FF], are:

- PSA updateable root of trust, such as Software isolation framework, protecting more trusted software from less trusted software, Generic services such as binding, initial attestation, generic crypto services, FW update validation.

- PSA immutable root of trust, for example Boot ROM, Root secrets and IDs, Isolation hardware, Security lifecycle management and enforcement. This component cannot be updated.
- Trusted Subsystems used by the PSA root of trust, such as security subsystems, trusted peripherals, SIM or SE, which include both hardware and software components are also in the scope of evaluation.

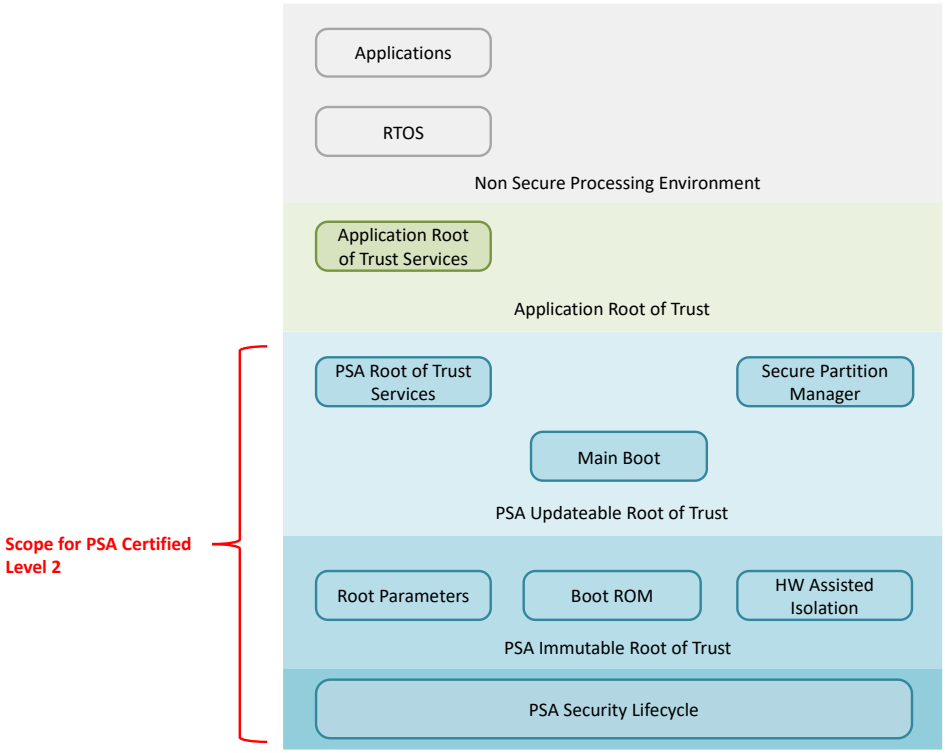

Figure 1: Scope of PSA Certified Level 2

The TOE consists of a secure boot and secure Firmware Update application and consists of a set of secure services running on a STM32L5 microcontroleur, 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 TF-M trusted firmware on the STM32L5 microcontroleur 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 application, the TF-M core, the secure crypto services, the secure storage service, the Internal Trusted Storage service and the secure initial attestation service.

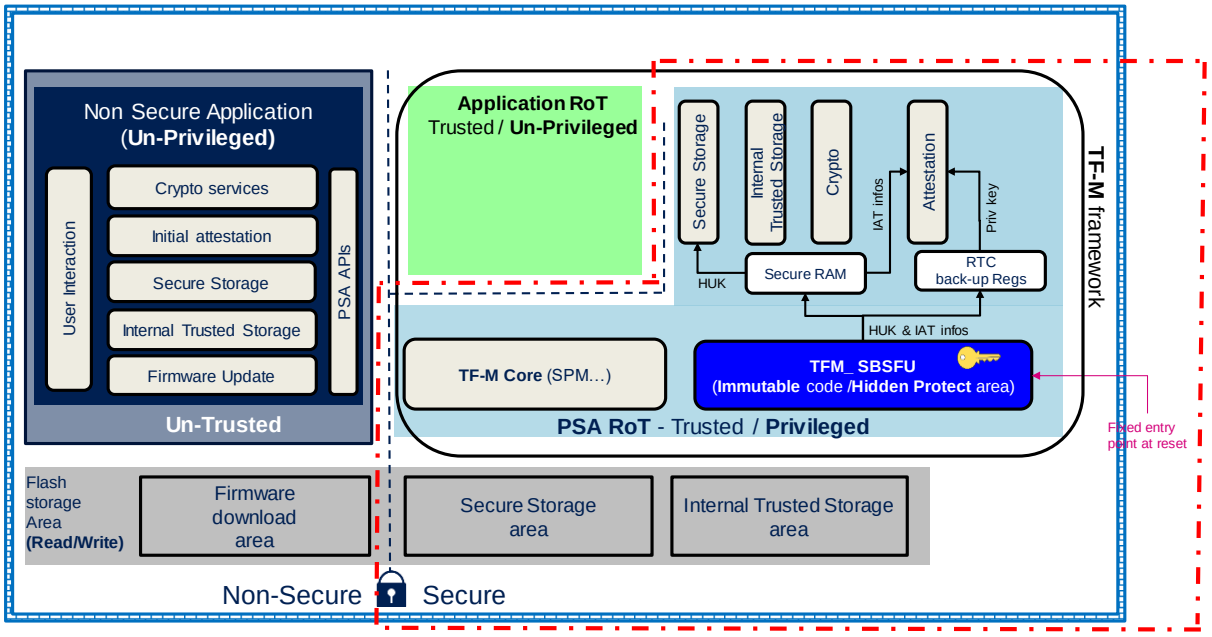

Figure 2: TOE scope

TFM framework uses SMT32L5 hardware security features (especially CM33 TZ, MPU and Hide Protect) to put in place 4 isolated domains:

- TFM_SBSFU application: secure privilege immutable code executed after reset managing the secure boot and the secure firmware functions. This code is provisioned with some assets (RSA 3072 public Key for TFM application authentication, IAT private key, IAT public key, IAT public SHA256), it will retrieve the HUK, it will generate the IAT Boot seed at each boot and it will share those information with TF-M trusted firmware via secure areas (secure SRAM and secure back-up registers). 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 application. This code puts in place TF-M Secure partitions andmanages 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 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.

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.

1.6.4 Usage and Major Security Features

The STM32L5 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 SESIPL3 certification:

- A Secure Processing Environment (SPE) isolated by STM32L5 hardware mechanisms (Turstzone) 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 bale 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 turst as STM32L5 hardware mechanisms allow to ensure that this code is immutable (non-volatile built-in flash memory protection (WRP) and Life cycle (RDP Level 1 or RDP Level 2) 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 yhe same STM32L5 hardware mechanisms.
- Support for Secure Storage, to protect in integrity and confidentiality sensitive assets such as the Hardware Unique Key (HUK), 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.
- Support for Internal Trusted Storage, to write data in a STM32L5 built in flash memory region that will be isolated from non-secure application or from non-secure/Unprivileged application thanks to the STM32L5 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 SPE 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.

1.6.5 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 fulfill its security requirements, the operational environment (technical or procedural) shall fulfil the following objectives:

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.

, as described in §4.2.4 “Security Measures” of [UM2745].

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.

, as described in §4.2.4 “Security Measures” of [UM2745].

TOE_PERSONALIZATION:

The integrator provisiones 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..

, as described in §4.2.4 “Security Measures” of [UM2745].

3 Security Requirements and Implementation

3.1 Security Assurance Requirements

The claimed assurance requirements package is: **SESIP3** as defined in [SESIP].

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 give 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-security@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 Cyber-Team will evaluate the potential vulnerability, by confirming the issue, analyze it and set a priority for resolving it.*
- *Solving: STM32 Cyber-Team 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 leadtime 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 SBSFU 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 Security Functional Requirements

The platform fulfills 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 1.3.1 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 STM32L5 HW.
- Implementation ID: contains SHA256 value computed on the immutable SW code part of the TOE (TFM_SBSFU 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 B (0x2000)
- STM32L552xx and STM32L562xx (0x472)

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.

Application note:

The identification must meet the attestation requirements of fields of [PSA L2 PP].

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 public key which is 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.

Informational:

This is the F.Attestation requirement of [PSA L2 PP].

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 application

- SW measurement: HASH of firmware controlled and computed by the TFM_SBSFU application
- Implementation ID: digest (SHA256) of the immutable SW code part of the TOE (TFM_SBSFU code binary data).
- Instance ID: digest (SHA256) of EAT public key (provisioned inside the TFM_SBSFU immuteable data region area).
- EAT public Key (provisioned inside the TFM_SBSFU immuteable data region area)
- Hardware ID: immutable STM32L5 HW version
- Life cycle: computed and verified by the TFM_SBSFU 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 backup registers)
- The non-secure application get the token and can answer back to the original requestor

Hardware ID, implementation ID, SW measurement and Instance ID informations 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 Update of Platform

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

Application note:

If the platform is only the hardware and PSA Immutable Root of Trust, only this SFR may be removed (use ~~strikethrough~~) as per SESIP requirements. If the platform includes firmware, that firmware must be updateable and this SFR must be kept and applied to that firmware.

Conformance rationale

TFM_SBSFU 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), authenticates the SPE image by digital signature (RSA-3072) 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 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 application.

TFM_SBSFU application handles the SPE and the NSPE images independently (multiple image boot). The 2 images are signed independently with different keys and they can be updated separately.

The current open source implementation of the TF-M RC2 does not implement data confidentiality during the update operation. However, confidentiality is considered to be trivially satisfied for the following reasons:

- The source code of the TOE is considered open source. Anyone is able to download it from the ST Microelectronics web page by request. As the source code is open source, not having confidentiality during the update operation is not considered an issue.
- 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.

3.2.5 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 secure state.

Application note:

The integrity of the root parameters: "Instance ID" and "Boot validation key (ROTPK)", as well as the integrity of the lifecycle state, should be included in these checks. (as per [PSA L2 PP] "4.1 Assets")

Conformance rationale

When the lifecycle state is set to RDP level1 (or level2) and the "Boot Lock" feature is set, the TOE boots on the immutable part of the internal flash which host TFM_SBSFU application code (secure boot) belonging to the TF-M firmware framework and STM32L5 HW ensures that data (such Instance ID, HUK…) 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 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-3072) validation.

TFM_SBSFU 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) as programmed in the immutable flash memory area or are computed (SAH256 of TFM_SBSFU_Boot application code, lifecycle state…) by TFM_SBSFU application (trusted immutable code executed in secure mode) using immutable parameters and HW immutable information (HW version…).

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.

3.2.6 Software Attacker Resistance: Isolation of Platform

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.

Application note:

"Platform" is software running in the PSA Secure Processing Environment, "application" is software running in the "Non-Secure Processing Environment". (as per [PSA L2 PP] "5.2 F.SOFTWARE_ISOLATION").

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 STM32L5 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 application code and to its associated assets) before executing the secure application.

3.2.7 Software Attacker Resistance: Isolation of Platform Parts

The platform provides isolation between platform parts, such that an attacker able to run code in **the Secure Processing Environment** can compromise neither the integrity and confidentiality of **PSA Root of Trust and other executable code (such as the Application Root of Trust) of the Secure Processing Environment** nor the provision of any other security 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 of STM32L5 in order to define a privilege domain (for PSA_RoT) and an unprivileged domain (for application_RoT) in the secure domain.

3.2.8 Secure Encrypted 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 lengths **128 bits**.

Application note:

The encryption must include integrity protection as per [PSA L2 PP] "5.3 F.SECURE_STORAGE".

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 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. The service 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 (128 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 Inprivileged 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

3.2.9 Cryptographic keyStore

The platform provides the application with a way to store **Cryptographic keys defined in Table 1** such that not even the application can compromise the **confidentiality** of this data. This data can be used for the cryptographic operations **defined in Table 1**.

Table 1 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.

Application note:

The keystore must include the Initial Attestation Key (IAK), a Hardware Unique Key (HUK) of at least **256 security** (**128 bits**), and if supported the boot encryption key.

Conformance rationale

IAT private key and HUK are programmed during product manufacturing stage in the immutable TFM_SBSFU data region (that is in secure/privileged domain and that will be hidden by TFM_SBSFU application thanks the the STM32L5 hardware hide Protect feature before executing the verified secure application). Before executing the SPE, the TFM_SBSFU application shares the IAT private key and HUK 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 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) 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.

3.2.10 Cryptographic Operation

The platform provides the application with **Cryptographic operations** functionality as specified in **Specification/Standard** for keylengths **Key size(s)**[1] and modes **Mode(s)**.

See values in Table 2.

Rationale

Table 2 Cryptographic operation iterations

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) IETF RFC 3610 (CCM)	128, 256 bits	ECB (HW) CBC (HW), CTR (HW), GCM (aead) (HW), GMAC (aead) (HW) , CCM (aead) (SW), CMAC (aead) (SW), CFB (HW)
RSA	Encryption Decryption Signature generation Signature verification	IETF RFC 8017 FIPS PUB 186-4	2048, 3072	SW implementation Encryption schemes: RSAES-OAEP, RSAES-PKCS1-v1_5, Signature scheme: RSASSA-PSS, RSASSA-PKCS1-v1_5, EMSA-PSS

[1] Application note: key sizes must be such that a security level of at least 128 bits is reached as per current state of the art. See keylength.com for a convenient summary.

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
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 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 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 application, the SPE execution will be started and the SPE will configure the dynamic security so that the initial attestion service of the SPE will be executed from the secure/privilege domain. The initial attestion 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 application or during the execution of the SPE application will be detected by the STM32L5 security HW mechanisms and will result in generating a HW reset or in executing an infinite loop in secure privileged domain.

3.2.12 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 1 or RDP Level 2.

When RDP 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, the product is locked and it is not possible to go to any other state.

3.2.13 Additional SFRs

3.2.13.1 Limited Physical Attacker Resistance

The platform detects or prevents attacks by an attacker with physical access before the attacker compromises **Field return of platform**.

Conformance rationale

The TOE in the certified configuration only allows to set the RDP to Level 1 or Level 2.

From Level 1, it is not possible to access the TOE contents with JTAG 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 interface with full debug capabilities).

From Level 2, debug connection is not possible and it is not possible to do any RDP regression to go to product virgin state. Product is locked in this configuration.

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

4 Mapping and sufficiency rationales

4.1 Assurance

The assurance activities defined in [PSA L2 EM] are fulfilled by SESIP3 level. In particular, the required source code review, vulnerability analysis and vulnerability analysis to an equivalent of 25 man days of the [PSA L2 EM] is applicable.

Assurance Class	**Assurance Families**	**Covered by**	**Rationale**
ASE: Security Target evaluation	ASE_INT.1 ST Introduction	Introduction	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	Security objectives for the operational environment	The table listing the objectives for the operational environment refers to the guidance documents.
	ASE_REQ.3 Listed Security requirements	Security Requirements	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	Security Functional Requirements	All SFRs are listed per definition, and for each SFR the implementation and verification is defined in Security Functional Requirements **Rationale.**
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 representation of the TSF to the SFRs	Implementation representation and mapping to SFRs as specified in "References"	The implementation representation can be mapped to the SFRs defined in section "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	Section "References".	Configuration items are properly identified.
	ALC_FLR.2 Flaw reporting procedures	Flaw reporting procedures	The flaw reporting and remediation procedure is described.
ATE: Tests	ATE_IND.1 Independent testing: conformance	Proof of functional conformance testing	A subset of the TSFI is independently tested by the evaluator.

AVA: Vulnerability Assessment	AVA_VAN.3 Focused vulnerability analysis	Evaluation activities carried out by the certification laboratory	The vulnerability analysis and associated test results required by SESIP3 and carried out by the evaluation testing.

4.2 Functionality

Functional Requirement	(detail)	Covered by SESIP	Rationale
F.INITIALIZATION		SFR "Secure Initialization of Platform"	Full coverage by direct translation. Corresponds to F.INITIALIZATION, providing secure boot features
F.SOFTWARE_ ISOLATION	Isolation between NSPE and SPE	SFR "Software Attacker Resistance: Isolation of Platform"	Corresponds to F.SOFTWARE_ISOLATION and partially covers the isolation between NSPE and SPE.
	Isolation between ARoT and RoT	SFR "Software Attacker Resistance: Isolation of Platform Parts"	Corresponds to F.SOFTWARE_ISOLATION and partially covers the isolation between ARoT and RoT.
F.SECURE_ STORAGE		SFR "Secure Encrypted Storage"	Full coverage. Note the application note explicitly requiring the encryption to also cover integrity. Corresponds to F.SECURE_STORAGE for application assets
		SFR "Cryptographic keyStore"	Corresponds to F.SECURE_STORAGE, which does not differentiate between crypto assets and general assets.
F.FIRMWARE_ UPDATE		SFR "Secure Update of Platform"	Corresponds to F.FIRMWARE_UPDATE for the platform part. Confindentiality of the update operation is not mandated in PSA L2 and not implemented in SESIP.
F.SECURE_STATE	Protect against abnormal situations	SFR "Software Attacker Resistance: Isolation of Platform"	Corresponds to F.SECURE_STATE providing protection against abonormal situations.
	Control access to its services	SFR "Software Attacker Resistance: Isolation of Platform" and the SESIP vulnerability analysis.	Corresponds to F.SECURE_STATE providing control access to its services.

	Enters secure state	SFR "Secure Initialization of Platform" and SESIP vulnerability analysis.	Corresponds to F.SECURE_STATE providing secure boot features
	Initialization error or software failure detection	SFR "Attestation of Platform State"	Corresponds to F.SECURE_STATE to provide an externally verifiable platform-level attestation, essential in PSA
		SFR "Field return of platform"	Corresponds to F.SECURE_STATE. It is not possible to recover user data using RDP regression.
		SFR "Limited Physical Attacker Resistance"	Covers the physical attacks to preserve the integrity of the lifecycle state.
F.CRYPTO		SFR "Cryptographic Operation"	Corresponds to F.CRYPTO to provide crypto operations.
		SFR "Cryptographic keyStore"	Corresponds to F.CRYPTO to provide secure storage of keys.
F.ATTESTATION	Reports on device identity	SFR "Verification of Platform Identity"	This SFR is mandatory for all SESIP platform (parts).
		SFR "Verification of Platform Instance Identity"	Dependency from "Genuine platform instantiation" that provides a unique identifier.
	… can be verified by remote entities.	SFR "Attestation of Platform Genuineness"	Corresponds to F.ATTESTATION for the platform's identity.
	Firmware measurements … can be verified by remote entities.		
	Runtime status of the device ... can be verified by remote entities.	SFR "Attested secure state of platform"	Corresponds to F.ATTESTATION to provide an externally verifiable platform-level attestation, essential in PSA
F.AUDIT	Maintains log of all significant security events and allows access to these logs to authorized users only	NA	The PSA implementation, including the chip, does not implement the generation of audit records. The STM32L5 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		NA	Not claimed in PSA because this feature is not accessible to the user.

5 References

5.1 ST defined

References listed in this section are used to satisfy the contents requirements for a SESIP certification.

5.1.1 Guidance

Identified in section 1.4 of this ST, including operational user guidance and preparative procedures to satisfy AGD_OPE.1 and AGD_PRE.1 requirements.

5.1.2 Other references

The following references are used to justify the sufficiency of assurance requirements for a SESIP3 certified solution.

[ALC_CI]	ALC CMC and CMS, L5_TFM ALC_CMC&CMS, v1.0 P5
[ALC_FLR]	Internal Flaw reporting procedures, V0.6
[ADV_FSP]	Full functional specification, L5_TFM_ADV_FSP_mapping, v1.6
[ADV_IMP]	Mapping of the code to the SFRs, L5_TFM_ADV_IMP_mapping, v1.3
[CODE]	Full source code, STM32Cube_FW_L5_V1.1.0 SW package, v1.1.0
[TST]	API compliance test log 1.1.0, V1.0

5.2 Profile defined

[PSA L2 EM]	PSA Certified™ Level 2 Evaluation Methodology, JSADEN0003, Version 1.1, dated 18/02/2020
[PSA L2 PP]	PSA Certified™ Level 2 Lightweight Protection Profile, JSADEN0002, Version 1.1, dated 18/02/2020
[SESIP GP]	GlobalPlatform Security Evaluation Standard for IoT Platforms (SESIP), version 1.0 Public Release, March 2020.
[SESIP PP]	SESIP profile for PSA L2 Chip, TrustCB-SESIP-PP-ARMPSAL2CHIP, TrustCB-SESIP-PP-PSAL2CHIP, version 1.4
[SESIP]	Security Evaluation Standard for IoT Platforms, Version 1.3

6 Document History

Version	Date	Comment	Author
0.1	25/09/2019	Initial draft	Brightsight B.V.
0.2	02/10/2019	Draft for SESIP3 to be shared with ST	Brightsight B.V.
0.3	21/10/2019	ST comments added	ST Microelectronics
0.4	29/10/2019	Draft for SESIP3 to be shared with ST	Brightsight B.V.
0.5	30/10/2019	ST comments added	ST Microelectronics
0.6	04/11/2019	Minor fixes on the document	Brightsight B.V.
0.11	17/04/2020	Updates and comments	Brightsight B.V.
1.0 P1	24/04/2020	ST updates and comment	ST Microelectronics
1.0 P2	07/05/2020	Remove claim to Profile	Brightsight B.V.
1.0 P3	13/05/2020	Updated Security Requirements and added “Limited Physical Attacker Resistance”	Brightsight B.V.
1.0 P4	26/05/2020	Claim to [ARM PSA L2 PP] Adapt to ST Template v2.1 from TrustCB	Brightsight B.V.
1.0 P5	28/05/2020	Udate according to new SESIP specs from GlobalPlatform	ST Microelectronics
1.0 P6	30/06/2020	Update according to Certifier feedback	ST Microelectronics
1.0 P7	01/07/2020	Look and feel update. Security guidance title and reference updated.	ST Microelectronics
1.0 P8	15/07/2020	Fix automatic references	ST Microelectronics
1.0 P9	30/07/2020	Update according to new version of the profile and minor changes.	ST Microelectronics
1.0 P10	04/08/2020	Minor fixes on the document	ST Microelectronics