STM32H573xx Security Target for STM32Trust TEE Secure Manager for PSA Certified™ Level 3 with SESIP Profile

Document information

This Security Target document is based on the GlobalPlatform® Security Evaluation Standard for IoT Platforms (SESIP) [GP_FST_070], version 1.1 (June 2021).

1 Introduction

This Security Target document describes the STM32H573 platform and the exact security properties of the platform that are evaluated against the *GlobalPlatform® Security Evaluation Standard for IoT Platforms (SESIP)* [GP_FST_070].

The platform is the STM32Trust TEE Secure Manager (STM32TRUSTEE-SM) composed of the STM32H573xx MCU and Secure Manager package.

The protection profile reference and conformance claims for this Security Target are described in Table 1.

Table 1. Protection profile reference and conformance claims

Reference	Value
Protection profile name	SESIP protection profile for PSA Certified™ Level 3 [JSADEN011]
Protection profile version	1.0
Package claim	Base PP SFRs. Optional SFRs (refer to Section 1.4.1 for details).
Assurance claim	Refer to Section 3.1

1.1 Security Target reference

This document: Technical note *STM32H573xx Security Target for STM32Trust TEE Secure Manager for PSA Certified™ Level 3 with SESIP Profile* (TN1475), STMicroelectronics, revision 4.

1.2 Platform reference

The platform (TOE) is certified by composition (layered composition model) using STM32H573xx MCU (TOE1) as the `Base` component and the Secure Manager package as the `Dependent` component.

Figure 1 presents the composition:

Figure 1. Platform composition

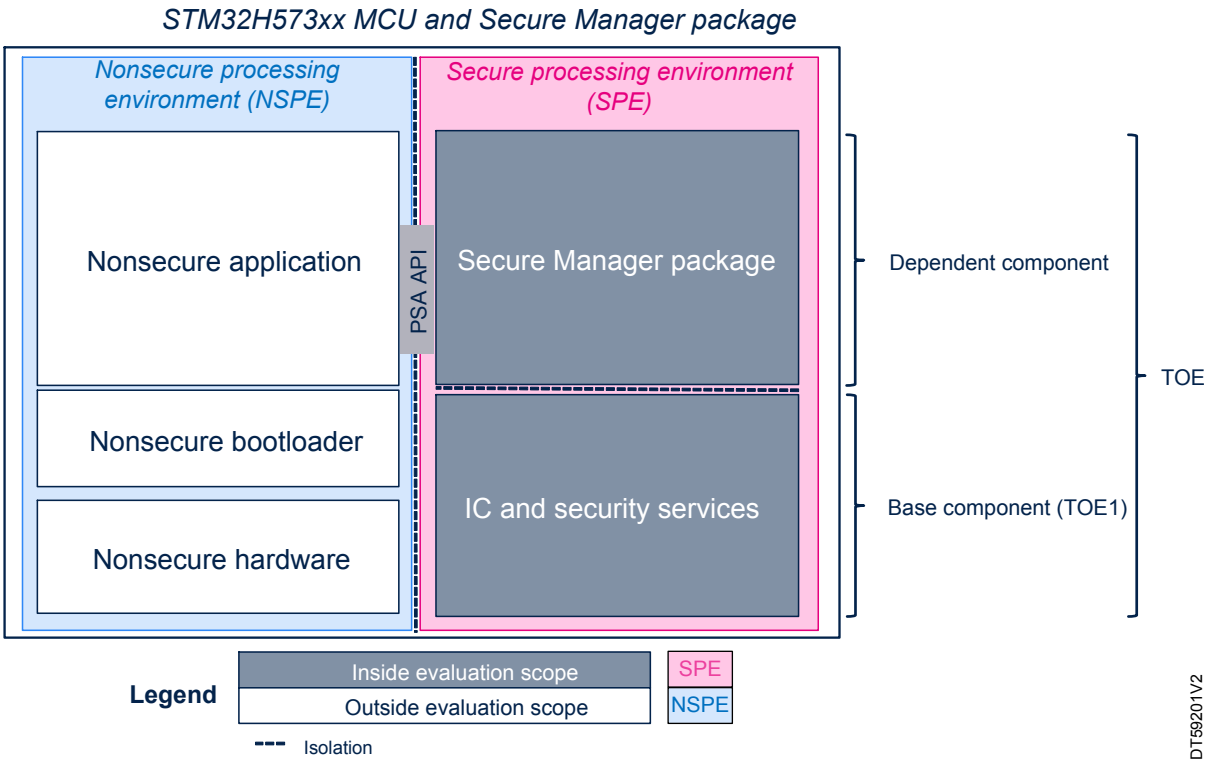


Table 2 describes the platform reference.

Table 2. Platform reference (hardware and software)

Reference	Value
Platform name and version	Secure Manager package: • Secure Manager, version 2.0.0 • SMuRoT, version 2.0.0 • SMiRoT, sha256 `0xc34fe5628acdf18c9a640b34d2a7b28dbef95f62dd54c64f08bb90c8942e5db1` STM32H573xx MCU • Refer to the composition element
Platform identification	STM32H573xx MCU version 1.3 with Secure Manager package version 2.0.0 (STM32Trust TEE Secure Manager)
Platform type	Microcontroller unit and Secure Manager package for IoT, industrial, or consumer applications.
Composition elements	STM32H573xx MCU (TOE1) composed of: IC: • STM32H573 (DieID= 0x484), version 1.3 (RevID= 0x1007) Security services: • STiRoT, version v2.3.0 • Debug Authentication, version v2.4.0 • Security library, version 2.14.0

1.3 Included guidance documents

The documents in Table 3 are included with the platform:

Table 3. Guidance documents (Secure Manager package)

Reference	Name	Version
[UM3238/SG_SMP]	User manual *STM32H573xx STM32TRUSTEE-SM security guidance for SESIP 3 Certification*	3
[TN1449/ST_SS]	Technical note *STM32H573xx Security Target for SESIP 3 Certification*	2
[UM3125/SG_SS]	User manual *STM32H573xx Security Guidance for SESIP 3 Certification*	2
[RM0481]	Reference manual *STM32H563/H573 and STM32H562 Arm®-based 32-bit MCUs*	1

1.4 Platform functional overview and description

1.4.1 Platform security features and scope

Security features

The platform is a microcontroller and security framework compliant with Arm® Platform Security Architecture (PSA) specifications for Cortex®-M (Armv8-M).

The microcontroller supports the following security features:

- Resource isolation using privilege mode and Armv8-M mainline security extension of Cortex®-M33, extended to securable I/Os, memories, and peripherals
- Boot entry: The platform makes it possible to select between ST immutable Root of Trust (in system flash memory) or proprietary boot entry (in user flash memory). In the Secure Manager case, ST immutable Root of Trust boot entry is selected.
- Security services: Security services are embedded in the system memory to manage the Root of Trust services. The immutable Root of Trust services take care of the platform security including secure boot, secure updates of the next boot level (uRoT: updatable Root of Trust), and secure debug control (debug reopening, regression control). Security services can be personalized for each OEM and personalization is done thanks to provisioning tools.
- Temporal isolation: Boot levels are isolated thanks to the HDPL (hide protect level) monotonic counter.
- Secure storage
- General purpose cryptographic acceleration
- New flexible life cycle scheme
- Active tampering and protection against temperature, voltage, and frequency attacks

The security framework (Secure Manager package) supports the following security features:

- Secure boot and firmware update
- PSA secure services to nonsecure application
  - Initial attestation
  - Cryptography
  - Internal trusted storage
  - Firmware update
- Multiple-tenant software IP protection
  - Sandboxed secure services (PSA isolation level 3)
- SFR verification of platform identity, verification of platform instance identity, and attestation of platform genuineness are supported through PSA initial attestation service.
- SFR residual information purging is also supported.

SFRs

The platform implements the SFRs listed under the platform support column in Table 4.

Find how to read different columns:

- PSA L3 PP request indicates if PSA L3 PP requests the SFR.
  - M: Mandatory
  - A: Additional
  - O: Optional
  - N or "": Not listed
- Platform support indicates if the platform supports the SFR
  - Y: Supported
  - N or "": Not supported
- STM32H573xx MCU dependency indicates if the SFR is dependent on STM32H573xx MCU (due to composition model)
  - Y: Dependency
  - N or "": No dependency

Table 4. Security functional requirements (SFRs)

SFR	PSA L3 PP request	Platform support	STM32H573xx MCU dependency
Base PP			
Verification of platform identity	M	Y	-
Verification of platform instance identity	M	Y	-
Attestation of platform Genuineness	M	Y	-
Secure initialization of platform	M	Y	-
Attestation of platform state	M	Y	-
Secure update of platform	M	Y	-
Physical attacker resistance	M	Y	Y
Software attacker resistance: Isolation of platform parts (between SPE and NSPE)	M	Y	-
Software attacker resistance: Isolation of platform parts (between PSA-RoT and application RoT services)	M	Y	-
Cryptographic operation	M	Y	-
Cryptographic random number generation	M	Y	-
Cryptographic key generation	M	Y	-
Cryptographic KeyStore	M	Y	-
Optional SFRs			
Software attacker resistance: Isolation of application parts (between each of the application Root of Trust services)	O	Y	-
Secure debugging	O	Y	Y
Secure encrypted storage (internal storage)	O	Y	-
Extra SFRs (on top of PSA L3 PP)			
Attestation of application genuineness	O	Y	-
Attestation of application state	O	Y	-
Factory reset of platform	O	Y	Y
Secure installation of application	O	Y	-
Secure update of application	O	Y	-
Software attacker resistance: Isolation of platform parts	O	Y	-
Residual information purging	O	Y	Y

The SFRs are not implemented using a secure enclave.

Platform scope

The platform consists of the STM32H573xx MCU and Secure Manager package.

The STM32H573 MCU is the SESIP-certified member of the STM32H5xx family of general-purpose microcontroller solution (MCU). It ensures superior cyber protection for cost- and power-conscious connected devices, also providing high-end core performance and peripheral integration.

The Secure Manager package is an updatable software composed of an updatable boot stage (SMuRoT) and security framework (Secure Manager) providing services to the application at runtime.

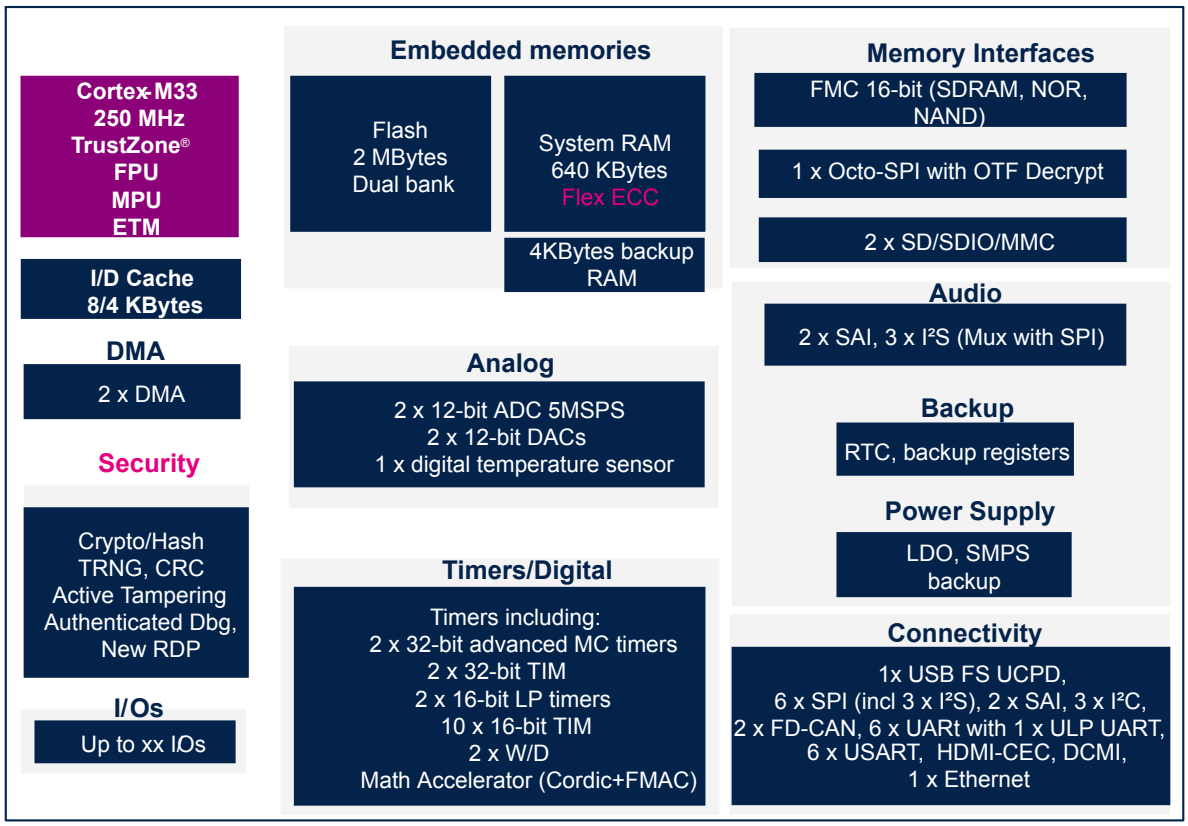
The platform intends to be used by integrators as a basis to develop an IoT platform by adding the required components, including their applications, as well as additional hardware components.

Physical scope

The physical scope of the platform is composed of the STM32H573xx MCU (refer to [RM0481] for details).

The STM32H573xx MCU overview is described in Figure 2:

Figure 2. STM32H573 block diagram



Note: Arm and TrustZone are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

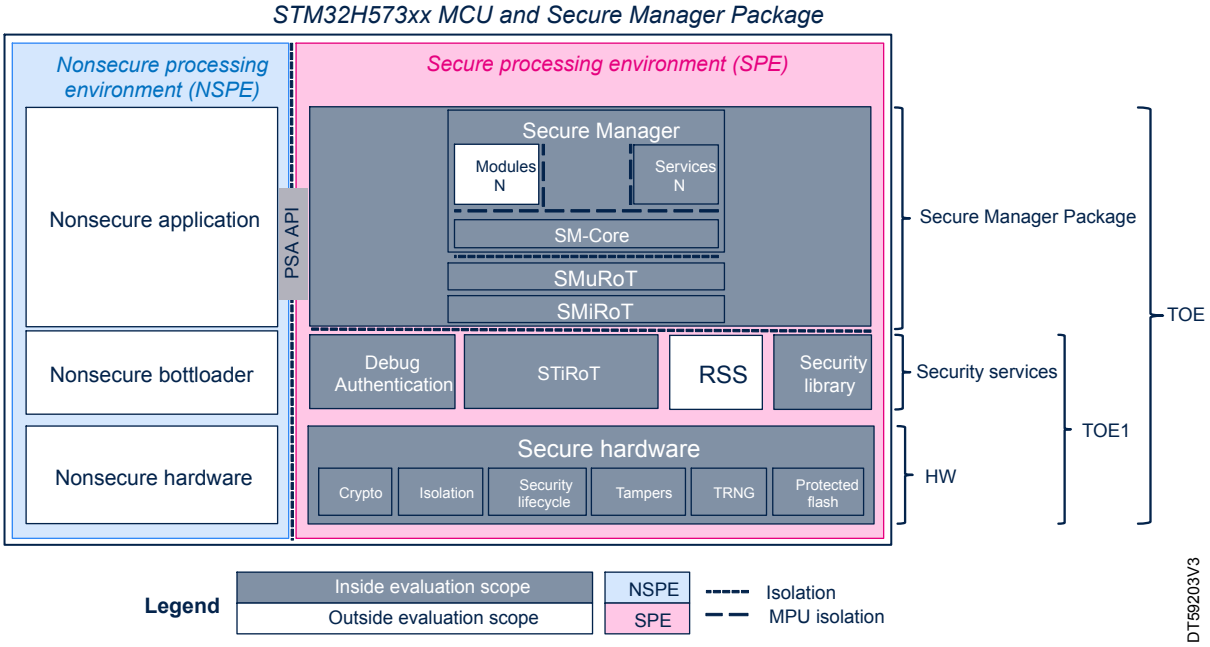
Logical scope

The logical scope of the platform is composed of the STM32H573xx security services and the Secure Manager package.

The physical (hardware) and logical (software) scope of the certification is described in Figure 3.

Note: The enabled first level-boot is SMiRoT, not STiRoT.

Figure 3. Detailed platform scope



Software components of the platform

Table 5 lists the software components and interfaces of the Secure Manager package.

Table 5. Software components and interfaces of the Secure Manager package

Component/ Interface	Description	Identification/ Version
SMiRoT	Immutable Root of Trust. It is responsible for the secure boot and the secure firmware update of the SMuRoT.	Refer to Section 1.2
SMuRoT	Updatable Root of Trust. It is responsible for the secure boot and the secure firmware update of the Secure Manager, secure modules, and NS application.	Refer to Section 1.2
Secure Manager	Software supplying secure services to NS application. It is composed of: • SM Core: Responsible for core services, especially partition management to schedule the secure services and modules • Secure services: Responsible for providing secure runtime services to the NS application, especially: – PSA crypto services – PSA ITS services – PSA attestation services – PSA firmware update services	Section 1.2
APIs	Refer to [SG_SMP] for API references	Refer to [SG_SMP]

Table 6 lists the software components and interfaces of the STM32H573xx MCU.

Table 6. Software components and interfaces of the STM32H573xx MCU

Component/ Interface	Description	Identification/ Version
STiRoT	Portion of immutable firmware that manages the secure boot and the secure firmware update of the next boot stage (SMuRoT) (code and/or related nonvolatile data) installed in the integrated user flash memory and option byte area	Refer to [ST_SS]
Security library	Portion of immutable firmware that manages the jump from RoT to the bootloader or from iRoT to the application	Refer to [ST_SS]
Debug authentication	Portion of immutable firmware in charge of debug reopening or regression meaning erasing memory contents.	Refer to [ST_SS]
APIs	Refer to [SG_SMP] for API references.	Refer to [SG_SMP]

Table 7 lists the hardware components and interfaces of the STM32H573xx MCU.

Table 7. Hardware components and interfaces of the STM32H573xx MCU

Component/ Interface	Description	Identification/ Version
STM32H573xx	STM32H573xx MCU integrated circuit	Refer to [ST_SS]
APIs	Refer to [SG_SMP] for API references	Refer to [SG_SMP]

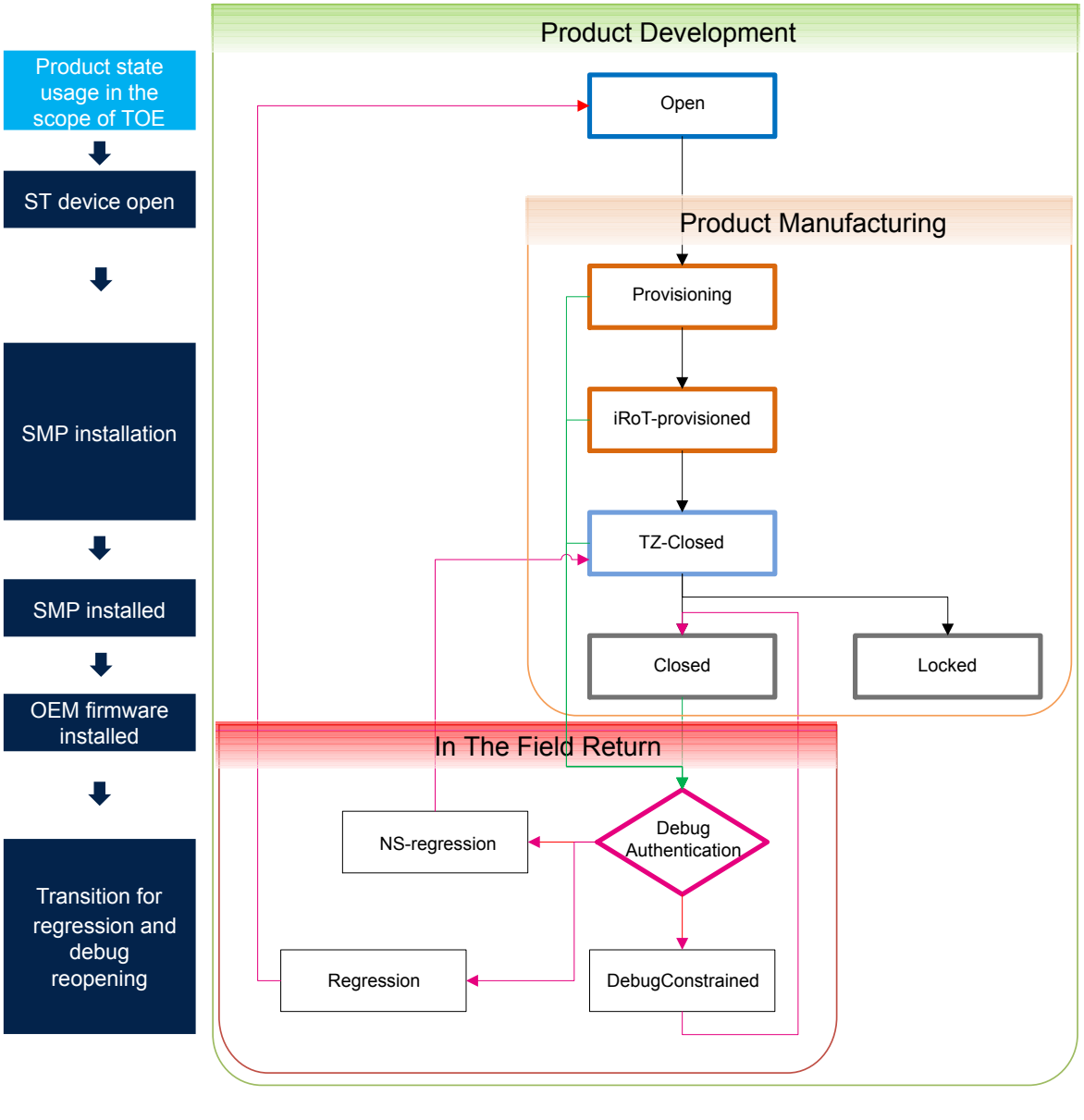
1.4.2 Life cycle

The life cycle of the platform under evaluation can be found in section 3.11 of [RM0481].

The details of life cycle usage in the scope of the Secure Manager package platform are described in [SG_SMP].

An overview of this live cycle is described in Figure 4:

Figure 4. Connected platform life cycle overview



Here are described typical product state transitions.

Product state: Open

ST delivers the STM32 device in the Open state.

ST delivers the Secure Manager package in a signed and encrypted format to the OEM.

OEM installs the Secure Manager package using a secure Loader delivered by STMicroelectronics.

After this installation, depending on customer choice, the STM32 device is either in the TZ-Closed, Closed, or Locked product state. In this state, the Secure Manager package code is loaded in secure user flash and is not accessible by the application.

Product state: TZ-Closed

OEM installs its OEM images using its Loader (or secure Loader delivered by ST).
After this OEM installation, depending on OEM choice, the STM32 device is in the Closed or Locked product state. OEM images are no longer accessible via the debugging interface.

Product state: Closed

In this state, OEM images are no longer accessible via the debug interface. OEM can reopen NSPE debug using the Debug Authentication method. Refer to [SG_SMP)].

Product state: Locked

In this state, OEM images are no longer accessible via the debug interface. OEM cannot reopen NSPE debug using the Debug Authentication method. Refer to [SG_SMP)].

For more details on firmware installation, refer to [SG_SMP)].
For more details on the Debug authentication method, refer to [SG_SMP)].

1.4.3 Use case

The platform intends to be used by an integrator as a SESIP level 3 compliant Root-of-Trust basis to develop a connected product by adding to it the required components. Such components include NS application, secure modules, as well as additional hardware components as required by the final product.
The environmental conditions under which the TOE can be securely used are defined below:

- **[any user]** The product may be physically accessed by an unknown or untrusted user, in an environment where access to the product cannot be sufficiently controlled or even in a more hostile environment.
- **[any code]** It cannot be excluded that the product executes code that is unknown to the product developer.

1.4.4 Required hardware, software, and firmware

On top of the Secure Manager package, the platform is supplied with:

- X-CUBE-SEC-M-H5 V2.0.0 Expansion Package that contains the Secure Manager package binary.
- STM32CubeProg tool used to program the Secure Manager package.

In addition, Python™ must be installed on the host.
To get the packages and tools versions to be used, refer to [UM3238].

2 Security objectives for the operational environment

2.1 Platform objectives for the operational environment

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

- The operating system or application code is expected to verify the correct version of all platform components that it depends on, as described in Section 1.2.
- The operating system or application code is expected to use the secure boot feature as described in [SG_SMP].
- The integrating environment is expected to configure the debug functionality as described in [SG_SMP] to meet the extra physical attacker resistance.

Table 8. Platform objectives for the operational environment

ID	Description	Reference
KEY MANAGEMENT	Cryptographic keys and certificates outside of the platform are subject to secure key management procedures.	Refer to [SG_SMP], section 4.2.4 “Security measures”
TRUSTED_USERS	Actors in charge of platform management, for instance for the signature of firmware updates, are trusted.	Refer to [SG_SMP], section 4.2.4 “Security measures”
UNIQUE_ID	The platform user must provide the integrity and uniqueness of the unique identification of the platform during the personalization stage.	Refer to [SG_SMP], section 4.2.4 “Security measures”

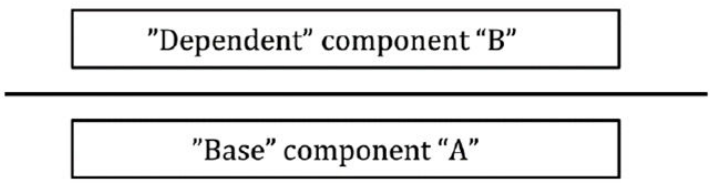
2.2 Inherited objectives for the operational environment

Composition model

The layered composition model is based on the `Base` and `Dependent` components. The principle is that certification of the platform composed of `Base` and `Dependent` components can reuse certification results from the `Base` component.

This model is described in Figure 5:

Figure 5. Layered composition model



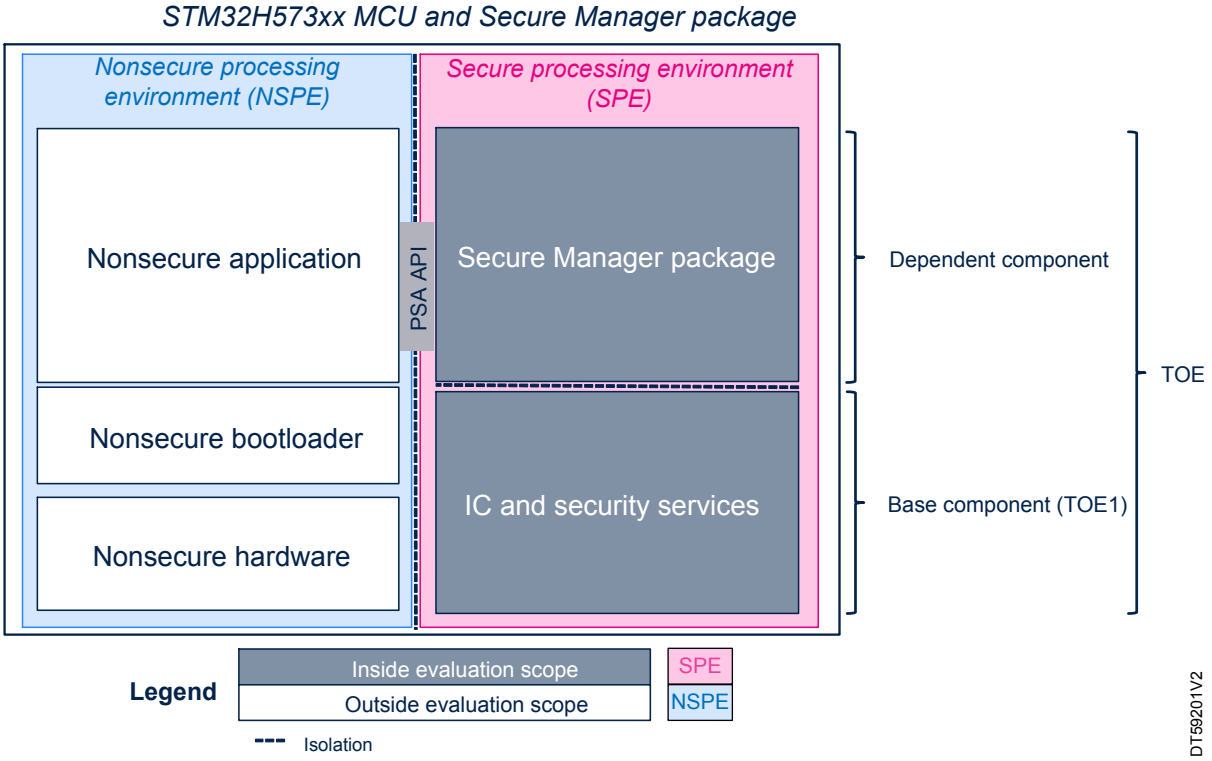
In our composition model, the platform (TOE) is composed of STM32H573xx MCU (TOE1) which is the `Base` component and the Secure Manager package, which is the `Dependent` component.

The STM32H573xx MCU is certified under the SESIP protection profile for secure MCUs and MPUs.

The platform is certified under the SESIP PSA L3 profile, using STM32H573xx MCU certification results via composition.

The composition view is described in Figure 6:

Figure 6. Composite view



Due to this composition, the platform must fulfill the operational environment objectives of STM32H573xx MCU.

Platform part: STM32H573xx MCU

The STM32H573xx MCU defines the objectives (defined in [ST_SS]) for its operational environment, that the platform (SMP and SMP operational environment) handles as follows:

- The operating system or application code is expected to verify the correct version of all platform components that it depends on, as described in [SG_SS].
  - The operational environment must verify the correct version of such components as described in [SG_SMP].
- The operating system or application code is expected to use the secure boot feature as described in [SG_SS] with STiRoT configuration
- In the case of using the debug authentication capabilities, the integrating environment is expected to configure the debug functionality as described in [SG_SS] section 3.3 to meet the extra physical attacker resistance.
  - The operational environment shall configure the debug functionality as described in [SG_SMP].

3 Security requirements and implementation

3.1 Security assurance requirements

The claimed assurance requirements package is SESIP3, as defined in chapter 4 of the *GlobalPlatform® Technology Security Evaluation Standard for IoT Platforms (SESIP)* [GP_FST_070].

3.2 Flaw reporting procedure (ALC_FLR.2)

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

3.3 Security functional requirements organization

3.4 Base PP security functional requirements

The platform fulfills the following security functional requirements:

3.4.1 Verification of platform identity

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

Conformance rationale:

Secure Manager

The verification of platform identity relies on the PSA initial attestation service.

This service supplies an API to get an authenticated token containing all the hardware and software component identifiers and versions.

The following token claim defines the platform software components:

- Implementation ID: Identify immutable PSA RoT

The following token claims define the platform software components:

- Software components:
  - A list of software components that represent all the software loaded by the PSA RoT
  - For each software component:
    - Measurement type
    - Measurement value
    - Version
    - Signer ID
    - Measurement description

3.4.2 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:

Secure Manager

The verification of platform identity relies on the PSA initial attestation service.

This service supplies an API to get an authenticated token.

This token contains a claim-named instance ID. The instance ID is the SHA-256 of the EAT public key, which is unique per chip.

3.4.3 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 ensures that the platform cannot be cloned or changed without detection.

Conformance rationale:

Secure Manager

The verification of platform identity relies on the PSA initial attestation service.

This service supplies an authenticated token containing the `Platform identity` and the `Platform Identity Instance`. The platform signs this token in a way that the verification entity can verify the signature to ensure that the `Platform Identity` and the `Platform Identity Instance` are valid.

The signature is based on the following algorithm: ECDSA P256 over SHA-256.

3.4.4 Secure initialization of platform

The platform ensures its integrity and authenticity during platform initialization. If the platform authenticity or integrity cannot be ensured, the platform goes to a state where no other operation, except optionally the secure update of the platform, can be performed.

Conformance rationale:

The secure initialization of the platform is ensured thanks to the Root of Trust implemented by SMiRoT, SMuRoT, and Secure Manager.

The platform is configured to boot in SMiRoT. For this, the unique boot entry (UBE) option (option bytes) is configured to boot in the system flash memory (SMiRoT).

The state described above is implemented as a system reset or a jump to the system bootloader. In case of a jump to the system bootloader, the secure update of the platform can be performed.

SMiRoT

In this configuration, after each reset the platform boots on SMiRoT.

SMiRoT then manages the secure boot of the SMuRoT installed inside the user flash by:

- Verifying SMuRoT integrity (before executing it), using the referenced SHA-256 stored in the platform
- Verifying SMuRoT authenticity using ECDSA P256 over SHA-256, using a public key stored in the platform. This authentication occurs each time the application is installed.

Once done, the SMiRoT jumps to the SMuRoT.

The STM32H573 product state is at least set to TZ-Closed.

If during the initialization process, an integrity or authenticity error is detected, the SMiRoT must either perform a system reset or a jump to the nonsecure loader application, which only allows the download of a new image version.

If during the initialization process, a security configuration error is detected, the system does not start to execute any application. It results in a system reset.

SMuRoT

When SMuRoT is executed:

- It verifies the static security configuration.
- It ensures the authenticity and integrity of:
  - NS application (using ECDSA P256 over SHA-256)
  - Secure Manager (using ECDSA P256 over SHA-256)
  - Secure modules
    - OEM modules signed (using ECDSA P256 over SHA-256)
    - Third-party modules with HSM (using AES GCM)
    - Third-party modules without HSM (using AES GCM)
    - Third-party modules can be on top signed with OEM key (using ECDSA P256 over SHA-256)
- It jumps to the Secure Manager

If during the initialization process, an integrity or authenticity error is detected, the SMuRoT must either perform a system reset or jump to the nonsecure loader application, which only allows the download of a new image version.

If during the initialization process, a security configuration error is detected, the system does not start to execute any application. It results in a system reset.

3.4.5 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:

Secure Manager

The attestation of the platform state relies on the PSA initial attestation service.

This service generates an authenticated token containing the platform state. The platform state is contained in the `Security Lifecycle` token claim.

At each boot, SMuRoT computes the platform state after the secure initialization of the platform is successfully verified.

3.4.6 Secure update of platform

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

Conformance rationale:

The platform implements the secure update of the platform as follows:

- The Secure Manager supplies platform update services to the NS application thanks to the PSA firmware update service
- At each boot, if a new secure update of a platform part can be required:
  - If required, SMiRoT updates SMuRoT
  - If required, SMuRoT updates the Secure Manager

SMiRoT and SMuRoT ensure integrity, authenticity, and confidentiality during this secure update procedure.

Secure Manager

Secure Manager implements the PSA firmware update service to update a new SMuRoT or Secure Manager image version. To do this, it relies on SMiRoT and SMuRoT services, executed at each boot.

SMiRoT

At each reset, SMiRoT is executed:

- It is executed in secure mode.
- It detects that there is a new SMuRoT image version installation request and manages it securely.
- It authenticates the image using ECDSA P256 over SHA-256, using the ST public key.
- It decrypts the image encryption key using ECIES-P256, using the ST private key.
- It decrypts the image using AES CTR 128 bits, using the ST key.

Once SMiRoT finishes the installation, it jumps to SMuRoT.

SMuRoT

At each reset, SMuRoT is executed after SMiRoT:

- It is executed in secure mode.
- It detects that there is a new Secure Manager image version installation request and manages it securely.
- It authenticates the image using ECDSA P256 over SHA-256, using the ST public key.
- It decrypts the image encryption key using ECIES-P256, using the ST private key.
- It decrypts the image using AES CTR 128 bits, using the ST key.

Once SMuRoT finishes the installation, it jumps to the Secure Manager.

Note: The secure updates of the NS application and secure modules are handled the same. This is described in the Secure update of application.

3.4.7 Physical attacker resistance

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

Conformance rationale:

The platform is configured in the TZ-Closed, Closed, or Locked product state. In TZ-Closed, JAG access to SPE is forbidden. In Closed and Locked states, JTAG access to SPE and NSPE is forbidden.

On top of that, the platform uses the following countermeasures against physical attacks:

- Tamper detection and response, such as automatic erasing of both SRAM and backup registers
- Hardware crypto engine SCA/DPA resistant
- Secure Debug port disabling
- Software mitigations to counter physical attacks, such as code duplication and flow control

SMiRoT

SMiRoT provides the following hardware countermeasures against physical attacks:

- Tamper detection and response, such as automatic SRAM erasing and software platform reset
- Hardware crypto engine SCA/DPA resistant
- Software mitigations to counter physical attacks, such as code duplication and flow control

SMuRoT

SMuRoT provides the following hardware countermeasures against physical attacks:

- Tamper detection and response, such as automatic SRAM erasing and software platform reset
- Hardware crypto engine SCA/DPA resistant
- Software mitigations to counter physical attacks, such as code duplication and flow control

Secure Manager

Secure Manager provides the following hardware countermeasures against physical attacks:

- Tamper detection and response, such as automatic SRAM erasing and backup registers
- Hardware crypto engine SCA/DPA resistant
- Software mitigations to counter physical attacks, such as code duplication and flow control

3.4.8 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 a code as an application on the platform cannot compromise any other claimed security functional requirements.

Conformance rationale:

For this SFR, the application means NS application (running on NSPE).

The platform ensures isolation of the platform (between SPE and NSPE) thanks to hide protection (HDP), TZ, and MPU isolation.

This is implemented by the different platform parts: SMiRoT, SMuRoT, and Secure Manager.

3.4.9 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 any other claimed security functional requirements.

Conformance rationale:

For this SFR, the application means a secure module (running on SPE).
The Secure Manager ensures isolation of the platform (between PSA-RoT and application Root of Trust services) thanks to full sandboxing of PSA-RoT and application Root of Trust. For this purpose, MPU_S is used.

3.4.10 Cryptographic operation

The platform provides the application with *operations in Table 9* functionality with *algorithms in Table 9* as specified in *specifications in Table 9* for key lengths *described in Table 9* and modes *described in Table 9*.

Conformance rationale:

Secure Manager

The platform (Secure Manager) supplies these cryptographic operations thanks to the PSA cryptography service.

Table 9. Secure Manager cryptographic operations

Operations	Algorithms	Specification	Key lengths	Modes
Encryption, decryption	AES	FIPS PUB 197 NIST SP800-38A	128, 256	ECB, CBC, CTR
Authenticated encryption or decryption		NIST SP800-38C NIST SP800-38D	128, 256	GCM, CCM
Cipher-based message authentication code		NIST SP800-38B IETF RFC 4493	128	CMAC
Cryptographic hash	SHA-2(1)	FIPS PUB 180-4	N/A	SHA-224, SHA-256 SHA2-384, SHA2-512
Asymmetric encryption	RSA	PKCS #1: RSA encryption V1.5	2048, 3072	RSAES-OAEP, PKCS1-v1_5
Signature with hashing Signature verification	RSA	IETF RFC 8017 FIPS PUB 186-4	2048, 3072	PKCS1-v1_5, PSS
	ECDSA	ANSI X9.62 IETF RFC 7027 FIPS PUB 186-4	Up to 521	List of supported ECCs is provided in the note(2). Refer to FIPS PUB 186-4 for details.
Key agreement	ECDH	ANSI X9.42 IETF rfc8236	Up to 521	

1. *These algorithms must not be used when manipulating sensitive information like keys.*
2. *Regarding ECC, the following elliptic curves are supported: secp224r1, secp256r1, secp384r1, secp521r1, secp256k1, bp256r1, bp384r1, and bp512r1.*

Note: secp192r1 and secp192k1 are also supported but are not advised.

These implementations use the STM32 crypto engines described in [RM0481]. Refer to sections 4.10, 33, 34, and 36. They are protected against side-channel and timing attacks.

Secure Manager

All algorithms are declared in the STM32H573xx MCU security target [ST_SS], except the following ones.
The AES-CMAC algorithm is implemented as a CMAC software layer using SAES hardware in ECB/CBC mode.
PKCS1-v1_5 is implemented as a software layer using the PKA hardware.

3.4.11 Cryptographic random number generation

The platform provides the application with a way based on *a live entropy source (analog) to generate random numbers as specified in [NIST_800-90B]*.

Conformance rationale:

Secure Manager

The platform supplies cryptographic random number generation thanks to the PSA cryptography service.
For this purpose, the Secure Manager uses hardware RNG IP, compliant with the NIST SP800-90B specification.

3.4.12 Cryptographic key generation

The platform provides the application with a way to generate cryptographic keys for use in cryptographic operations in *Table 10* as specified in *specifications in Table 10* for key lengths *described in Table 10*.

Conformance rationale:

Secure Manager

The platform supplies the cryptographic key generation through the PSA cryptography API.
The hardware TRNG of the chip is used for key generation.

Table 10. Platform cryptographic key generation operations

Operations	Algorithms	Specifications	Key lengths	Modes
Key generation	AES	FIPS PUB 197 NIST SP800-38A	128, 256	All: ECB, CBC, CTR, GCM, CCM, CMAC
	HMAC	FIPS PUB 198-1	≥ 1 bit	-
	ECDH	ANSI X9.42	192, 224, 256, 384, 512, 521 bits	EC curves (hardware) secp192r1, secp224r1, secp256r1, secp384r1, secp521r1, secp192k1, secp256k1, bp256r1, bp384r1, bp512r1
	RSA	FIPS PUB 186-4	2048, 3072	-

3.4.13 Cryptographic KeyStore

The platform provides the application with a way to store *secret keys* such that not even the application can compromise the *authenticity, integrity, and confidentiality* of this data. This data can be used for cryptographic operations *(encryption, decryption, and authenticated encryption/decryption)*.

Conformance rationale:

Secure Manager

The platform (Secure Manager) supplies the cryptographic KeyStore through the PSA cryptography service.
The OEM securely provisions the secret keys. The procedure for the OEM securely to provision secret keys is described in [SG_SMP].
Once provisioned, the secret keys are accessible for the NS application through the PSA cryptography API, using an opaque key mechanism (the key is addressed by identifier and not by key value). In this way, authenticity, integrity, and data confidentiality are ensured.
ST also securely provisions some keys (DUA initial attestation and DUA user). The DUA user key is accessible through the PSA cryptography service and the DUA initial attestation key is accessible through the PSA initial attestation service.

3.5 Additional security functional requirements

Not applicable

3.6 Optional security functional requirements

3.6.1 Software attacker resistance: Isolation of application parts (between each of the application Root of Trust services)

The platform provides isolation between parts of the application, such that an attacker able to run code as one of the Application Root of Trust Secure Partitions cannot compromise the integrity and confidentiality of the other application parts.

Conformance rationale:

Note: The Application Root of Trust Secure Partitions corresponds to a secure module.

Note: The vulnerable application part is the one that can compromise the protected application part.

Note: The protected application part is the one we want to protect.

The Secure Manager (SM-Core) ensures the isolation of application parts (between each secure module) thanks to the full sandboxing of the secure modules. For this purpose, MPU_S is used. SM-Core is executed in a secure privileged module and the secure module is executed in secure unprivileged mode. In this way, only SM-Core can configure MPU_S.

The vulnerable application parts and protected application parts are described in Table 11.

The security mechanism to protect the protected application part from the vulnerable application part is also described.

Table 11. Isolation of application parts

Protected application part	Vulnerable application part	Protection mechanism
Secure module N	Secure module M (M#N)	MPU_S. SM-Core configures MPU_S in a way that a secure module cannot access resources, such as code, data, and peripherals of another secure module. The secure module is executed in a secure unprivileged area. On top of that MPU_S read, write, and execute attributes are used.

3.6.2 Secure debugging

The platform only provides *a JTAG or SWD debugger interface* authenticated as specified in *[PSA_ADAC]* with debug functionality.

The platform ensures that all the data stored by the application, except none, is made unavailable.

Conformance rationale:

The platform provides secure debugging access to authenticated users as defined in the PSA ADAC specification [PSA_ADAC].

A host initiates the authentication protocol and communicates to the device using the JTAG interface.

The authentication protocol is based on a challenge-response scheme based on ECDSA P256 with SHA-256.

The protection of assets is based on HDP (hide protection) levels, which allow the protection of some assets depending on the product state:

- HDPL1: The code and data of the security services are protected.
- HDPL2: The code and data of SMuRoT are protected.
- HDPL3S: The code and data of the Secure Manager and modules are protected.
- HDPL3NS: The code and data of the NS application are protected.

Once the platform is installed, the platform is executed in TZ-Closed, Closed, or Locked product state, based on OEM configuration.

The Secure Debug authentication process only allows opening HDPL3NS, which means that debug and regression are only allowed on the NS application.

In the TZ-Closed state, the debug interface is only available to access the NS application, without the usage of the Secure debug authentication protocol.

In the Closed state, the debug interface is not available. The Secure Debug authentication protocol allows opening the debug interface only for the NS application. OEM can configure the platform to allow:

- NS application debug or NS application regression. In this case, the product is set to the TZ-Closed state
- System regression. In this case, the product is set to the Open state

In the Locked state, the debug interface is not accessible. Debug reopening is not possible even with the Secure Debug authentication protocol.

The security services fully implement Secure Debug, especially RSSDA.

Security services

RSSDA is stored in system flash and is responsible for Secure Debug. It detects a debug authenticated request. It verifies the authenticity using a public key stored in the device. It verifies that the request is in line with a secure debug configuration (SOC_MASK) stored in the device.

Debug opening of SPE is forbidden. Debug opening of NSPE can be enabled or disabled based on the configuration parameter of the owner application.

RSSDA executes the secure debug request, which can be:

- NS application debug opening (up to next reset)

3.6.3 Secure encrypted storage (internal storage)

The platform ensures that all data stored by the application, except for *the data stored in NSPE*, are encrypted as specified in *the AES-CBC specification* with a platform instance unique key of key length *256-bit*.

Conformance rationale:

Secure Manager

The secure encrypted storage relies on the PSA internal trusted storage (ITS) service.

Data is accessible only through a call to PSA ITS services.

Data is encrypted thanks to:

- A chip-specific unique key of 256 bits (DHUK) derived from HUK
- AES-CBC algorithm

3.7 Other security functional requirements

The platform fulfills the following security functional requirements:

3.7.1 Attestation of application genuineness

The platform provides an attestation of the application, in a way that ensures that the application has not been cloned or changed without detection.

Conformance rationale:

Secure Manager

The attestation of application genuineness relies on the PSA initial attestation service.

This service supplies a token containing the `Software components` claim. The platform signs this token in a way that the verification entity can verify the signature and ensure that the `Software components` are valid.

`Software components` claims are either the NS application or the secure modules.

The signature is based on the following algorithm: ECDSA P256 over SHA-256.

3.7.2 Attestation of application state

The platform provides an attestation of the state of the application.

Conformance rationale:

Secure Manager

The attestation of the Application state relies on the PSA initial attestation service.

The Application state is the same as the platform state and is contained in the initial attestation token, in the `Security Lifecycle` field.

Refer to the Attestation of platform state for details.

3.7.3 Factory reset of platform

The platform can be reset to the state in which it exists when the product embedding the platform is delivered to the user, before any personal user data, user credentials, or user configuration is present on the platform.

Conformance rationale:

The factory reset of the platform is ensured thanks to the debug authentication procedure defined by [PSA_ADAC]. This procedure allows a full regression of the product, erasing all application code and data. This full regression sets the platform to the Open state. From this state, the product manufacturer must install the Secure Manager package and the application, as described in [SG_SMP]. Then user data can be reinstalled.

The Debug Authentication procedure is accessible via the platform debug port and implemented by the Debug Authentication firmware, part of security services.

3.7.4 Secure installation of application

The application can be installed in the field such that the integrity, authenticity, *and confidentiality* of the application is maintained.

Conformance rationale:

Note: The application can be either an NS application or a secure module.

The secure installation of application is performed the same way as the secure update of application. This is why it is described in the Secure update of application.

3.7.5 Secure update of application

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

Conformance rationale:

Note: The application can be either an NS application or a secure module.

The secure update of application is performed the same way as the secure update of the platform. This is why it is described in the Secure update of platform.

On top of that, the following specificities apply.

SMuRoT

For NS application:

- SMuRoT authenticates the image using ECDSA over ECC curve 256p1, using the OEM public key.
- SMuRoT decrypts the image encryption key using ECIES-P256, using the OEM private key.
- SMuRoT decrypts the image using AES CTR 128 bits, using the OEM key.

For the secure modules:

- SMuRoT authenticates and decrypts the image.
- The authentication and decryption algorithm depends on the module type as defined in Table 12.

Note: Either the OEM who develops the product or a third-party can own the secure module.

Table 12. Secure module crypto schemes

Secure module type/UC	Function	Algorithm	Key owner
OEM module installation	Image Authentication Image Encryption Image encryption key asymmetric encryption	ECDSA over ECC curve 256p1 AES-CTR 128 bits ECIES-P256	OEM OEM OEM
Third-party module installation • without HSM	Image Authentication Image Encryption Image encryption key asymmetric encryption	AES GCM 128 bits AES-CTR 128 bits ECIES-P256/KDF2/AES GCM 128 bits	Third-party Third-party ST
Third-party module installation • with HSM	Image Authentication Image Encryption Image encryption key asymmetric encryption	AES GCM 128 bits AES-CTR 128 bits ECIES-P256/KDF2/AES GCM 128 bits	Third-party Third-party ST
Third-party module installation • without HSM • with OEM authentication	Image Authentication Image Encryption Image encryption key asymmetric encryption Image Authentication	AES GCM 128 bits AES-CTR 128 bits ECIES-P256/KDF2/AES GCM 128-bit ECDSA over ECC curve 256p1	Third-party Third-party ST OEM
Third-party module installation • with HSM • with OEM authentication	Image Authentication Image Encryption Image encryption key asymmetric encryption Image Authentication	AES GCM 128 bits AES-CTR 128 bits ECIES-P256/KDF2/AES GCM 128-bit ECDSA over ECC curve 256p1	Third-party Third-party ST OEM

3.7.6 Software attacker resistance: Isolation of platform parts

The platform provides isolation between platform parts, such that an attacker able to run code in *vulnerable platform parts* can compromise neither the confidentiality and integrity of *protected platform parts* nor the provision of any other security functional requirements.

Conformance rationale:

Note:
- *The vulnerable platform part is the one that can compromise the protected platform part.*
- *The protected platform part is the one we want to protect.*

The platform parts are security services, SMiRoT, SMuRoT, SM-Core, and SM services, such as the PSA cryptography service. Isolation of platform parts is ensured thanks to HDP hide protection and MPU_S as described in Table 13.

The vulnerable and protected platform parts are described in Table 13. The mechanism to protect the protected platform part from the vulnerable application part is also described.

Table 13. Isolation of platform parts

Protected platform part	Vulnerable platform part	Protection mechanism
SMiRoT	SMuRoT, Secure Manager	Hide protection (HDP-L1).
Security services	SMuRoT, Secure Manager	Hide protection (HDP-L1). Security services are executed at boot before SMuRoT and Secure Manager execution. Security services use a hide protection (HDP-L1) hardware mechanism to hide its code before jumping to SMuRoT.
SMuRoT	Secure Manager	Hide protection (HDP-L2).

Protected platform part	Vulnerable platform part	Protection mechanism
		SMuRoT is executed at boot before Secure Manager execution. SMuRoT uses a hide protection (HDP-L2) hardware mechanism to hide its code before jumping to the Secure Manager.
SM-Core	SM Service	Memory protection unit (MPU_S), PRIVCFGR. The SM-Core is executed in secure privilege mode. It configures MPU_S in a way that the SM Service is executed in secure unprivileged mode. In this way, SM Service cannot compromise SM-Core.
SM Service N	SM Service M (M#N)	Memory protection unit (MPU_S). The SM-Core configures MPU_S in a way that the SM Service cannot access resources, such as code, data, and peripherals of another SM Service. The SM Service is executed in a secure unprivileged area. On top of that MPU_S read, write, and execute attributes are used.

3.7.7 Residual information purging

The platform ensures that *all data (SRAM) used by the platform*, with the exception of *none*, is erased using the method specified in the *below section* before the memory is used by the platform or application again and before an attacker can access it.

Conformance rationale:

Secure Manager

The SRAM associated with a secure service is erased when the process terminates and is overwritten with random data.

Additionally, the Secure Manager buffers (PSA buffers) are zeroized at creation (malloc) and when freed. Also, the PSA crypto service initializes buffers with random values. Furthermore, in the event of tampering, the hardware resets the SRAM2 used for the cryptographic service.

SMiRoT

The SMiRoT erases all its SRAM areas before jumping to the Secure Manager.

The SMiRoT clears all its allocated SRAM by simply writing 0x0 on each allocated SRAM address.

The SMiRoT erases all its SRAM areas in case of tamper detection.

SMuRoT

The SMuRoT erases all its SRAM areas before jumping to the Secure Manager.

The SMuRoT clears all its allocated SRAM by simply writing 0x0 on each allocated SRAM address.

The SMuRoT erases all its SRAM areas in case of tamper detection.

Security services

Security services erase all their SRAM areas before jumping to the application.

Security services clear all their allocated SRAM by simply writing `0x0` on each allocated SRAM address.

Security services erase all their SRAM areas in case of tamper detection.

4 Mapping and Sufficiency Rationales

4.1 SESIP3 sufficiency

ASE: Security Target evaluation	ASE_INT.1 ST introduction	Section 1	The ST reference is in the title, the TOE reference in the *Platform Reference*, and the TOE overview and description in *Platform Functional Overview and Description*.
	ASE_OBJ.1 security requirements for the operational environment	Section 2	For the objectives for the operational environment in *Security objectives for the operational environment*, refer to the guidance documents.
	ASE_REQ.3 listed security requirements	Section 3.4 to Section 3.7	All SFRs in this ST are taken from [GP_FST_070]. *Verification of Platform Identity* and *Secure Update of Platform* are included.
	ASE_TSS.1 TOE summary specification	Section 3	All SFRs are listed per definition, and for each SFR the implementation and verification are defined in *Security functional requirements*.
ADV: Development	ADV_FSP.4 complete functional specification	Section 1.3 and material provided to the evaluator	The platform evaluator determines whether the provided evidence is suitable to meet the requirement.
	ADV_IMP.3 complete mapping of the implementation representation of the TSF to the SFRs	Material provided to the evaluator	The platform evaluator determines whether the provided evidence is suitable to meet the requirement.
AGD: Guidance documents	AGD_OPE.1 operational user guidance	Section 1.3	The platform evaluator determines whether the provided evidence is suitable to meet the requirement.
	AGD_PRE.1 preparative procedures	Section 1.3	The platform evaluator determines whether the provided evidence is suitable to meet the requirement.
ALC: Life cycle support	ALC_CMC.1 labeling of the TOE	Section 1.3	The platform evaluator determines whether the provided evidence is suitable to meet the requirement.
	ALC_CMS.1 TOE CM coverage	Section 5	The platform evaluator determines whether the provided evidence is suitable to meet the requirement.
	ALC_FLR.2 flaw reporting procedures	Section 3.2	The flaw reporting and the remediation procedure are described.
ATE: Tests	ATE_IND.1 independent testing: Conformance	Material provided to the evaluator	The platform evaluator determines whether the provided evidence is suitable to meet the requirement.
AVA_VAN.3	AVA_VAN.3 focused vulnerability analysis	NA The platform evaluator performs a vulnerability analysis to ascertain the presence of potential vulnerabilities.	The platform evaluator performs penetration testing, to confirm that the potential vulnerabilities cannot be exploited in the operational environment for the TOE. The platform evaluator performs a penetration test assuming a potential enhanced-basic attack.

4.2 PSA security function mapping

Table 14. Functionality mapping and Sufficiency Rationales

PSA security function	Covered by SESIP SFR	Rationale
F.INITIALIZATION	Secure initialization	Full coverage
F.SOFTWARE_ ISOLATION	Software attacker resistance: Isolation of platform (between SPE and NSPE)	Full coverage
	Software attacker resistance: Isolation of platform (between PSA-RoT and application Root of Trust services)	Full coverage
	Software attacker resistance: Isolation of application parts (between each of the application Root of Trust services)	Full coverage
F.SECURE_ STORAGE	Secure encrypted storage (internal storage)	Full coverage
	Secure storage (internal storage)	No
	Secure external storage	No
F.FIRMWARE_ UPDATE	Secure update of platform	Full coverage
F.SECURE_STATE	Software attacker resistance: Isolation of platform (between SPE and NSPE)	Full coverage
	Software attacker resistance: Isolation of platform (between PSA-RoT and application Root of Trust services)	Full coverage
	Secure initialization	Full coverage
	Secure update of platform	Full coverage
F.CRYPTO	Cryptographic operation	Full coverage
	Cryptographic KeyStore	Full coverage
	Cryptographic random number	Full coverage
	Cryptographic key generation	Full coverage
F.ATTESTATION	Verification of platform identity	Full coverage
	Verification of platform instance identity	Full coverage
	Attestation of platform Genuineness	Full coverage
	Attestation of platform state	Full coverage
F.AUDIT	Audit log generation and storage	Full coverage
F.DEBUG	Secure debugging	Full coverage
F.PHYSICAL	Physical attacker resistance	Full coverage
Other SFRs	Attestation of application genuineness	Full coverage
	Attestation of application state	Full coverage
	Factory reset of platform	Full coverage
	Secure installation of application	Full coverage
	Secure update of application	Full coverage
	Software attacker resistance: Isolation of platform parts	Full coverage
	Residual information purging	Full coverage

5 Reference documents

Table 15. Reference documents

Reference	Definition
Evaluation documents	
[GP_FST_070]	*GlobalPlatform® Security Evaluation Standard for IoT Platforms (SESIP), version 1.1 (June 2021)*
[GP_SPE_150]	*GlobalPlatform® SESIP Protection Profile for Secure MCUs and MPUs, version 1.0 (Oct 2021)*
[JSADEN011]	*Arm® SESIP Profile for PSA Certified™ Level 3, version 1.0 REL, release number 02 (Nov. 2022)*
Development documents	
[UM3125/SG_SS]	*STM32H573xx Security Guidance for SESIP 3 Certification, version 2*
[UM3238/SG_SMP]	*STM32H573xx STM32TRUSTEE-SM security guidance for SESIP 3 Certification, version 3*
Standards	
[PSA_ADAC]	*Authenticated Debug Access Control, version 1.0, Arm Limited, DEN0101*
[NIST_800_90B]	*NIST, Special Publication 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation, January 2018, https://doi.org/10.6028/NIST.SP.800-90B*

6 Glossary

Table 16. **Glossary**

Term	Definition
Application	Used in SESIP to refer to the components that are out of the scope of the evaluation.
Debug Authentication	Immutable firmware responsible for debug reopening and regression.
(Secure) module	Secure Service that can be installed independently of the Secure Manager image.
Nonsecure processing environment (NSPE)	The processing environment that hosts the nonsecure 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.
NS application	Application executed in a nonsecure processing environment.
Platform	Used in SESIP to refer to the components that are in the scope of the evaluation. It is a synonym for a connected platform.
Product	Used by SESIP as a synonym for connected product.
PSA Root of Trust	In platform security architecture security Model v1.0 the PSA-defined combination of the immutable platform Root of Trust and the updateable platform Root of Trust, are considered to be the most trusted security component on the device.
Secure Manager	Secure firmware executed in the secure processing environment, responsible for supplying secure services to the NS application and the secure modules.
Secure Manager - Core	Part of the Secure Manager responsible for supplying secure Core services such as secure services isolation and scheduling.
Secure Manager package	Firmware composed of Secure Manager and SMuRoT.
Secure module	The secure module is executed in a secure processing environment. It is executed on a partition under the control of SM-Core. NS application uses the functions of the secure module through the PSA API.
Secure processing environment (SPE)	The processing environment that hosts the PSA-RoT, and any application RoT services.
(Secure) Service	Secure Service is executed in a secure processing environment. It is executed on a partition under the control of SM-Core. NS application uses the functions of the secure services through the PSA API.
Secure library	Immutable firmware supplying secure services at runtime.
Security services	Immutable secure services are mainly composed of Debug Authentication, STiRoT, and the secure library.
SMiRoT	Secure Manager immutable firmware is responsible for the first stage of secure boot and secure firmware update.
SMuRoT	Secure Manager updatable firmware is responsible for the second stage of secure boot and secure firmware update.
STiRoT	ST immutable firmware is responsible for the first stage of secure boot and secure firmware update.
STM32Trust TEE Secure Manager	ST security solution composed of the STM32H5 MCU and the Secure Manager package

7 Abbreviations

Table 17. **Abbreviations**

Term	Definition
ADAC	Authenticated Debug Access Protocol
AES	Advanced Encryption Standard
API	Application Programmable Interface
CM	Counter Measures
CTR	Counter
DHUK	Derived Hardware Unique Key
DPA	Differential Power Analysis
EAT	Entity Attestation Token
ECC	Elliptic-curve Cryptography
ECDSA	Elliptic-curve Digital Signature Algorithm
ECIES	Elliptic-curve Integrated Encryption Scheme
FW	Firmware
GP	Global platform
HW	Hardware
IC	Integrated Circuit
ID	Identifier
IoT	Internet of Things
JTAG	Joint Test Action Group
MCU	Microcontroller Unit
MPU	Memory Protection Unit
MPU	Microprocessor Unit
NIST	National Institute of Standards and Technology
NS	Nonsecure
NSPE	Nonsecure Processing Environment
OEM	Original Equipment Manufacturer
PP	Protection Profile
PSA	Platform Security Architecture
PSA-RoT	PSA Root of Trust
PSIRT	Product Security Incident Report Team
RSS	Root Secure Services
RSSDA	Root Secure Services Debug Authentication
SAU	Secure Attribution Unit
SCA	Side Channel Attack
SECBB	Security Block Based
SECWM	Security Watermark
SESIP	Security Evaluation Standard for IoT platforms
SFI	Secure Firmware Installation
SFR	Secure Functional Requirement

Term	Definition
SM	Secure Manager
SMAK	Secure Manager access Kit
SMDK	Secure Manager Development Kit
SMEK	Secure Manager Evolution kit
SMP	Secure Manager package
SPE	Secure Processing Environment
SSFI	Secure ST Firmware Installation
ST	Security Target
iRoT	Immutable Root of Trust
UBE	Unique Boot Entry
uRoT	Updatable Root of Trust
SW	Software
TOE	Target Of Evaluation
TZ	TrustZone®
WRP	Write Protection

Revision history

Table 18. **Document revision history**

Date	Revision	Changes
06-Feb-2024	1	Initial release.
05-Dec-2024	2	Added Section 1.4.4: Required hardware, software, and firmware. Updated: • Secure Manager and STuRoT versions in Table 2. Platform reference (hardware and software) • UM3238/SG_SMP version and name in Table 3. Guidance documents (Secure Manager package) • Security features in Section 1.4.1: Platform security features and scope • Conformance rationale: and STiRoT in Section 3.4.4: Secure initialization of platform • Section 3.6.1: Software attacker resistance: Isolation of application parts (between each of the application Root of Trust services) • UM3238/SG_SMP name in Table 15. Reference documents
22-Jul-2025	3	Update for Secure Manager version 2.0.0 • Table 2. Platform reference (hardware and software) • Table 3. Guidance documents (Secure Manager package) • Table 4. Security functional requirements (SFRs) • Figure 3. Detailed platform scope • Section 1.4.4: Required hardware, software, and firmware • Section 3.4.7: Physical attacker resistance • Table 13. Isolation of platform parts
29-Jul-2025	4	Updated: • Section 1.1: Security Target reference • Table 2. Platform reference (hardware and software)

Contents

1 **Introduction** 2
1.1 Security Target reference 2
1.2 Platform reference 2
1.3 Included guidance documents 4
1.4 Platform functional overview and description 4
1.4.1 Platform security features and scope 4
1.4.2 Life cycle 8
1.4.3 Use case 10
1.4.4 Required hardware, software, and firmware 10
2 **Security objectives for the operational environment** 11
2.1 Platform objectives for the operational environment 11
2.2 Inherited objectives for the operational environment 11
3 **Security requirements and implementation** 13
3.1 Security assurance requirements 13
3.2 Flaw reporting procedure (ALC_FLR.2) 13
3.3 Security functional requirements organization 13
3.4 Base PP security functional requirements 13
3.4.1 Verification of platform identity 13
3.4.2 Verification of platform instance identity 13
3.4.3 Attestation of platform genuineness 14
3.4.4 Secure initialization of platform 14
3.4.5 Attestation of platform state 15
3.4.6 Secure update of platform 15
3.4.7 Physical attacker resistance 16
3.4.8 Software attacker resistance: Isolation of platform (between SPE and NSPE) 16
3.4.9 Software attacker resistance: Isolation of platform (between PSA-RoT and application Root of Trust services) 16
3.4.10 Cryptographic operation 17
3.4.11 Cryptographic random number generation 17
3.4.12 Cryptographic key generation 18
3.4.13 Cryptographic KeyStore 18
3.5 Additional security functional requirements 18
3.6 Optional security functional requirements 19
3.6.1 Software attacker resistance: Isolation of application parts (between each of the application Root of Trust services) 19
3.6.2 Secure debugging 19

3.6.3 Secure encrypted storage (internal storage) . . . 20
3.7 Other security functional requirements . . . 20
3.7.1 Attestation of application genuineness . . . 20
3.7.2 Attestation of application state . . . 20
3.7.3 Factory reset of platform . . . 21
3.7.4 Secure installation of application . . . 21
3.7.5 Secure update of application . . . 21
3.7.6 Software attacker resistance: Isolation of platform parts . . . 22
3.7.7 Residual information purging . . . 23
4 Mapping and Sufficiency Rationales . . . 24
4.1 SESIP3 sufficiency . . . 24
4.2 PSA security function mapping . . . 25
5 Reference documents . . . 26
6 Glossary . . . 27
7 Abbreviations . . . 28
Revision history . . . 30
List of tables . . . 33
List of figures . . . 34

List of tables

Table 1. Protection profile reference and conformance claims . . . 2
Table 2. Platform reference (hardware and software) . . . 3
Table 3. Guidance documents (Secure Manager package) . . . 4
Table 4. Security functional requirements (SFRs) . . . 5
Table 5. Software components and interfaces of the Secure Manager package . . . 8
Table 6. Software components and interfaces of the STM32H573xx MCU . . . 8
Table 7. Hardware components and interfaces of the STM32H573xx MCU . . . 8
Table 8. Platform objectives for the operational environment . . . 11
Table 9. Secure Manager cryptographic operations . . . 17
Table 10. Platform cryptographic key generation operations . . . 18
Table 11. Isolation of application parts . . . 19
Table 12. Secure module crypto schemes . . . 22
Table 13. Isolation of platform parts . . . 22
Table 14. Functionality mapping and Sufficiency Rationales . . . 25
Table 15. Reference documents . . . 26
Table 16. Glossary . . . 27
Table 17. Abbreviations . . . 28
Table 18. Document revision history . . . 30

List of figures

Figure 1. Platform composition . . . 3
Figure 2. STM32H573 block diagram . . . 6
Figure 3. Detailed platform scope . . . 7
Figure 4. Connected platform life cycle overview . . . 9
Figure 5. Layered composition model . . . 11
Figure 6. Composite view . . . 12

IMPORTANT NOTICE – READ CAREFULLY

STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, enhancements, modifications, and improvements to ST products and/or to this document at any time without notice.

In the event of any conflict between the provisions of this document and the provisions of any contractual arrangement in force between the purchasers and ST, the provisions of such contractual arrangement shall prevail.

The purchasers should obtain the latest relevant information on ST products before placing orders. ST products are sold pursuant to ST’s terms and conditions of sale in place at the time of order acknowledgment.

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

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

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

If the purchasers identify an ST product that meets their functional and performance requirements but that is not designated for the purchasers' market segment, the purchasers shall contact ST for more information.

ST and the ST logo are trademarks of ST. For additional information about ST trademarks, refer to www.st.com/trademarks. All other product or service names are the property of their respective owners.

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