Technical note

STM32H7Sxx Security Target for Security Services

Document information

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

1 Introduction

The Security Target describes the STM32H7Sxx Platform and the exact security properties of the Platform that are evaluated against [1] (in Chapter 3 “Security Requirements and Implementation”) that a potential consumer can rely upon the product upholding if they fulfill the objectives for the environment (in Chapter 2 “Security Objectives for the Operational Environment”).

The protection profile reference and conformance claims for this Security Target are described below.

Table 1. Protection Profile Reference and Conformance Claims for TOE_WITH_STIROT Configuration

Reference	Value
Protection profile name	SESIP protection profile for secure MCUs and MPUs [2]
Protection profile version	1.0
Package claim	Base PP, security services, software isolation, hardware protections
Assurance claim	Refer to Section 3.1.

Table 2. Protection Profile Reference and Conformance Claims for TOE_WITHOUT_STIROT and TOE_WITH_STIROT configurations

Reference	Value
Protection Profile name	SESIP Profile for PSA Certified RoT Component Level 3 [3]
Protection Profile version	1.0
Package claim	Base PP
Assurance claim	Refer to Section 3.1.

1.1 Security Target Reference

This document: Technical note *STM32H7Sxx Security Target for Security Services* revision 3, STMicroelectronics.

1.2 Platform Reference

Table 3. Platform Reference

Reference	Value
Platform name and version	Integrated circuit: STM32H7S DieID= 0x485 + 1.2 (RevID= 0x1003) with full cryptographic configuration=(0x5200281C:bits8;16;17;18=0) Immutable firmware versions: • Configuration TOE_WITH_STIROT: – STiROT version: v1.1.0 – Debug authentication version: v1.0.0 – Security library version: 1.1.0 – RSS version: v1.1.0 • Configuration TOE_WITHOUT_STIROT: – Debug authentication version: v1.0.0 – Security library version: 1.1.0 – RSS version: v1.1.0
Platform identification	STM32H7Sxx
Platform type	Microcontroller platform, with Security Services as immutable firmware, for IoT, industrial, or consumer applications.

1.3 Included guidance documents

The following documents are included with the platform:

Table 4. Guidance documents

Document	Name	Reference
User manual	STM32H7Sxx security guidance for SESIP 3 Certification	UM3290
Product reference manual	STM32H7Rx/Sx Arm®-based 32-bit MCUs	RM0477

1.4 Platform functional overview and description

The STM32H7Sxx microcontroller is the SESIP-certified member of the STM32H7Rx/7Sx Arm® Cortex®-M7 high performance microcontrollers (MCU).

It ensures a common and optimized IoT platform protection on feature-rich and multidisciplined devices, while also providing external memory protection with ST's Memory Crypto engine (Encrypt/decrypt on the fly).

The platform consists of a single Arm® Cortex®-M7 at 600 MHz, with features such as STiRoT, Secure boot, Bootflash (user flash), Crypto acceleration, and internal and external memory protections.

1.4.1 Platform security features and scope

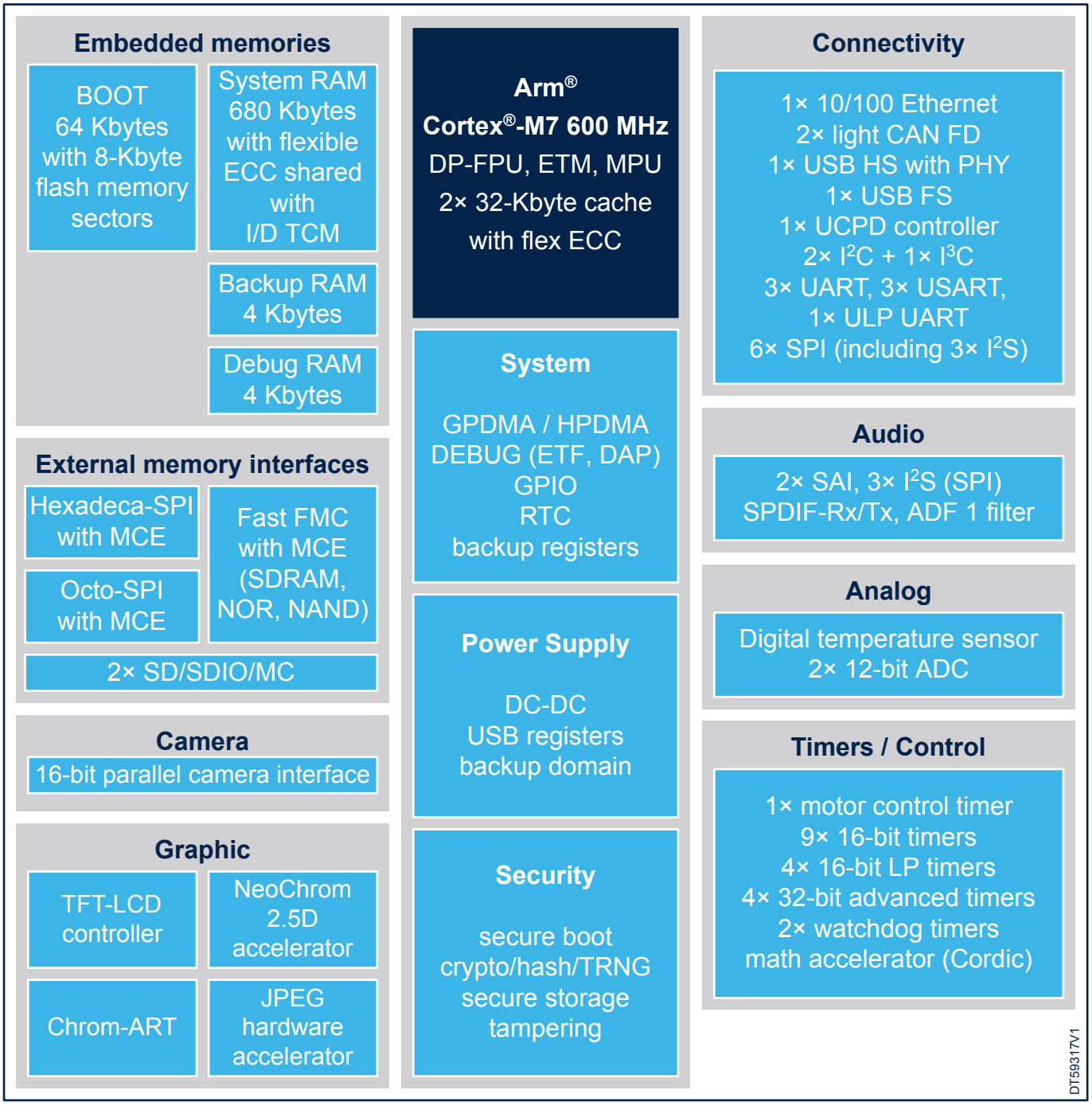
The STM32H7Sxx microcontroller is designed with a comprehensive set of security features, some of them based on the standard Arm® technology. Those features include:

- 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)
- Security Services: Security Services are embedded in the system memory to manage the Root of Trust services. Immutable Root of Trust services take care of 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 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

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

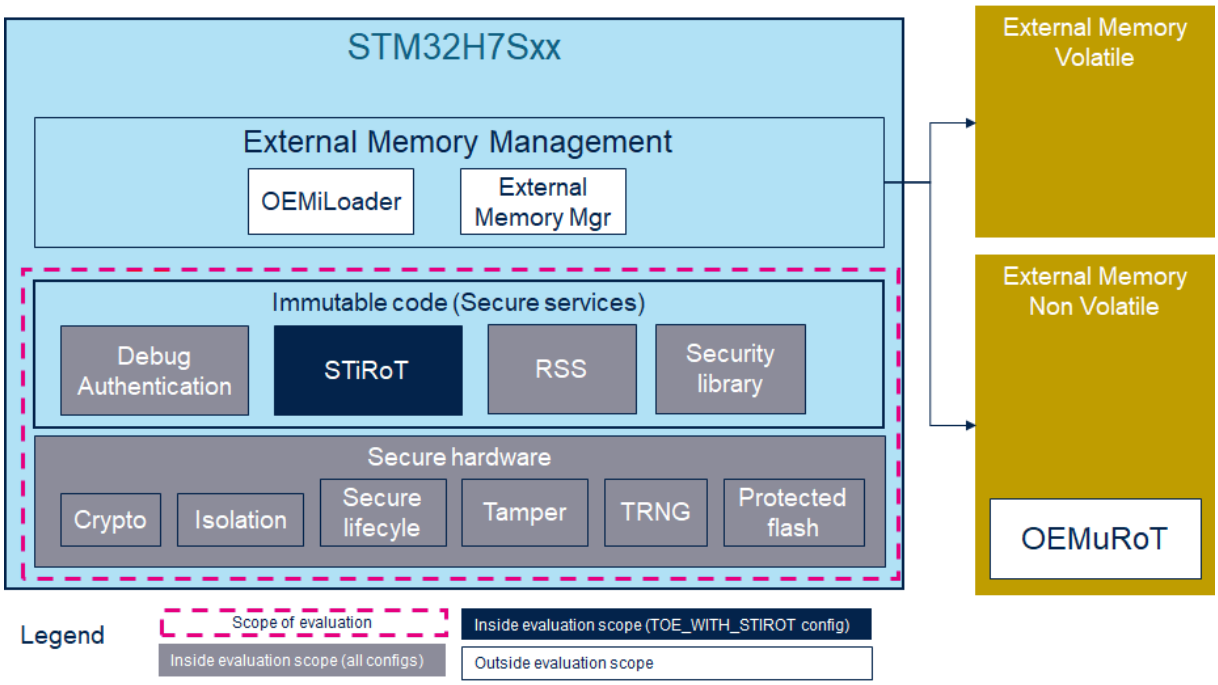
An overview of the STM32H7Sxx microcontroller is shown in the block diagram below.

Figure 1. STM32H7Sxx block diagram



The figure below details the hardware and firmware in the scope of the certification.

Figure 2. Detailed TOE scope



The physical scope of the TOE is the STM32H7Sxx integrated circuit, identified as defined in Section 1.2.

The hardware interfaces of the TOE are listed in section 4.2 of *STM32H7Sxx security guidance for SESIP 3 Certification* (UM3290). Refer to [4].

The logical scope of the TOE is defined in Table 5. Any additional firmware, OS, or application software stored on the platform is not in the scope of this evaluation.

Table 5. Software components and interfaces of the TOE

Component/ Interface	Description	Identification/ version
STiRoT	The portion of immutable firmware that manages the secure boot and the secure firmware update of the application (code and/or related nonvolatile data) installed in the integrated user flash and option byte area	1.1.0
Security library	The portion of immutable firmware that manages the jump from iROT to BL or from iROT to the application	1.1.0
Debug authentication	The portion of immutable firmware in charge of secure debug reopening or secure regression (meaning erasing memory content)	1.0.0
RSS	The portion of immutable firmware in charge of boot selection (to launch STiROT, Debug Authentication, bootloader, ...).	1.1.0
APIs	Refer to [5] to get the description of the APIs.	3

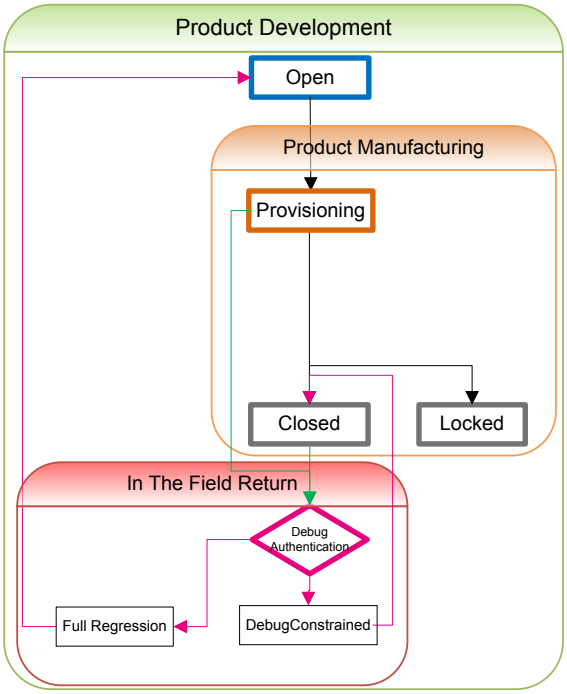
No additional nonplatform hardware, software, or firmware is required for the correct functioning of the security claims described in this document.

1.4.2 Life cycle

The life cycle of the platform under evaluation can be found in section 4.9 of the reference manual [5].

A graphical representation of it is provided here:

Figure 3. Connected platform life cycle overview



Note that some integrators might decide not to implement in their product a full Root of Trust firmware (for example a PSA-RoT).

In both TOE-certified configurations, the product state is set at least to *closed* or *locked* and the debug authentication service allowing the various regressions is always available unless the product state is *locked*.

For more details, refer to section 3.2 of [4] *STM32H7Sxx security guidance for SESIP 3 Certification* (UM3290).

1.4.3 Use case

The TOE is intended 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 a Root of Trust software layer, an operating system with connectivity, as well as additional hardware components as required by the final product.

As the TOE is certified in two different TOE configurations, the integrator might need to add its own Root of Trust implementation in user flash when using the TOE_WITHOUT_STIROT configuration. When using the TOE_WITH_STIROT configuration, TOE supports the following additional SFRs:

- Secure initialization of the platform
- Secure installation of the application
- Secure update of the application
- Secure storage
- Secure encrypted storage

The environmental conditions that have an impact on the security functional requirements implemented by the TOE in both configurations are listed below.

- **[Any user]** The product might 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 will execute code that is unknown to the product developer.

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 3.1 of [4].
- The operating system or application code is expected to make use of the secure boot feature as described in section 3.3 of [4].
- In the case of using the debug authentication capabilities, the integrating environment is expected to configure the debug functionality as described in section 3.3 of [4] to meet the extra physical attacker resistance inherited objectives for the operational environment.

The platform does not include platform parts previously evaluated under any SESIP certification scheme.

2.2 Inherited objectives for the operational environment

The Platform does not include Platform parts that have previously been evaluated under any SESIP certification scheme.

3 Security requirements and implementation

3.1 Security assurance requirements

The claimed assurance requirements package is SESIP Assurance Level 3 (SESIP3), as defined in Chapter 4 of GlobalPlatform® Technology Security Evaluation Standard for IoT Platforms (SESIP) [1].

3.1.1 Flaw reporting procedure (ALC_FLR.2)

Due to the TOE type (MCU hardware with immutable firmware), the SFR *Secure update of platform* is not applicable since updates of the TOE are impossible.

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

3.2 Common SFRs to every configuration

3.2.1 Base PP security functional requirements

The platform fulfills the following security functional requirements:

Secure Debugging

The platform only provides a debugger JTAG or SWD interface authenticated as specified in the PSA ADAC specification [6] with debug functionality.

The platform ensures that all data stored by the application, except those directly managed by the application, is made unavailable.

Conformance rationale:

The current SFR is only available if the integrator activates the Debug authentication capability on the platform, as explained within section 4.2.1 of [4].

When TOE is in the life cycle state “Closed”, the debug connection cannot be used. However, a trusted user can send to the platform via the debug port a signed debug certificate to open a debug session. This certificate is verified before granting debug reopening.

Debug authentication firmware stored in the platform immutable NVM (system flash) is responsible for secure debug reopening.

The platform runs the debug authentication firmware when detecting a debug reopening request. This firmware is responsible for the debug certificate verification.

Security of data is ensured by the HDPL mechanism, guaranteeing that when Debug is opened for HDPL=X, all data in lower levels protected by HDPL cannot be accessed. Typically, if Debug Authentication Control allows the debug reopening of the HDPL3, as soon as HDPL3 is reached the debugger becomes available, but all code and data protected by HDPL mechanisms (HDPL0,1,2) are not accessible. The debug reopening strategy is under OEM responsibility and depends on the product configuration and the certificate chain management (distribution).

3.2.2 Package “Security Services” security functional requirements

Cryptographic operation

The platform provides the application with side channel-resistant cryptographic operations such as encryption, decryption, authentication, and signature functionality with a list of algorithms specified in Table 6. TOE cryptographic operations versus key lengths and modes.

Conformance rationale:

The platform provides applications with the following side channel-resistant cryptographic algorithms, modes of operation and minimum/maximum key size. For more details, refer to section 4.8.1 in [5].

Some of those algorithms are used by STiRoT.

Table 6. TOE cryptographic operations

Operations	Algorithm	Specification	Key lengths	Modes
Encryption, decryption	AES[1]	FIPS PUB 197 NIST SP800-38A	128, 256 bits	ECB, CBC, CTR
Authenticated encryption or decryption		NIST SP800-38C NIST SP800-38D		GCM, CCM
Cipher-based message authentication code		NIST SP800-38D		GMAC
Protected modular exponentiation (signature, decryption, key agreement...)	RSA[2]	IETF RFC 8017 NIST SP800-56B FIPS PUB 186-4	Up to 4096 bits	RSA 2048, 3072, 4096
Signature	ECDSA	ANSI X9.62 IETF RFC 7027 FIPS PUB 186-4 SEC 1, SEC 2[3]	Up to 640 bits	Nist: P256, P384, P521 Brainpool: bp256r1, bp384r1, bp512r1 SEC 2[1]: secp256k1, secp256r1, secp384r1, secp521r1
ECC scalar multiplication (public key generation, key agreement, shared secret generation...)	ECDH ECIES	ANSI X9.42 ANSI X9.63 FIPS PUB 186-4 SEC 1, SEC 2[3]		

1. *AES algorithm with key sizes of 128 and 256 bits (and not DES/TDES) can run accelerated with side-channel attack resistance in SAES peripheral*
2. *Other operations not written in this table (like RSA CRT exponentiation or ECDSA signature verification) are not protected against side-channel attacks.*
3. *Standards for efficient cryptography: SEC1, SEC2*

The platform also provides useful cryptographic operations without special side-channel attack resistance. They are listed in the table below. It is recommended not to use these algorithms directly from the hardware peripherals when manipulating sensitive information.

Table 7. TOE cryptographic operations

Operations	Algorithm	Specification	Key lengths	Modes
Cryptographic hash	SHA-2	FIPS PUB 180-4	NA	SHA2-224, SHA2-256, SHA2-384, SHA2-512

3.2.3 Package “hardware protections” security functional requirements

The platform fulfills the following security functional requirements:

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 provides the following hardware countermeasures against physical attacks:

- Tampers:
  - Device Tamper detection: The platform offers user hardware features that detect tampers on the device embedding the platform.
  - System Tamper detection: STiROT configures the platform to detect tampers on the system's internal sensitive settings. On system tamper detection, STiRoT resets the platform.
  - Internal memories, whatever is on VM and NVM (including user flash and system flash). On memory tamper detection, STiRoT resets the platform.
    - Hardware Crypto engine SCA/DPA resistant.
    - Security Services implement software countermeasures such as:
  - Random jitter in the execution flow
  - Systematic verification of sensitive hardware security features activation after programming.
  - Control execution flow that prevents any sensitive security function bypass.
    - Debug: The platform debug port (JTAG/SWD) is closed and only the user having credentials provided by the platform integrator can reopen the debug port.

3.2.4 Additional security functional requirements

Verification of platform instance identity

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

Conformance rationale:

In addition to the platform identification and version mentioned in Section 3.3.1, the platform provides a Unique Device ID per chip (refer to section 67.1 of [5]).

Factory reset of platform

The platform can be reset to the state in which it exists when the composite 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 platform provides a service called regression supported by the debug authentication firmware stored in the system flash. This regression service erases all application code and data in nonvolatile memory (user flash). In addition, a platform power off and battery removal erase the nonvolatile memories. The regression service sets the platform product state to *OPEN*, which is the product state of the platform when it is delivered to the user. Refer to section 4.2.2 in [4], “JTAG interface” subsection, for details.

Regression service is accessible via the platform debug port (SWD/JTAG).

3.3 SFRs for TOE_WITH_STIROT configuration

3.3.1 Base PP Security Functional Requirements

Verification of platform Identity

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

Conformance rationale:

The platform referred to in Section 1.2 provides the following unique identifications:

- Integrated circuit hardware revision DieID and DieID, at `0x5C001000` and `0x5C001002` addresses:

```
Die ID: 485, meaning STM32H7Sxx
```

Address	Value (halfword)
0x5C001000	0xX485

Revision ID: v1.2

Address	Value (halfword)
0x5C001002	0x1003

Product configuration: Crypto (All crypto active

Address	Value
0x5200281C	Bit 8 (SAES, CRYP, PKA available): 0b0 Bit 16,17,18 (MCE available): 0b000

- Immutable firmware versions:

STiROT: v1.1.0

Address	Value (word)
0x1FF1FDCC	0xFF010100

Debug Authentication: v1.0.0

Address	Value (word)
0x1FF1FDC8	0xFF010000

Security Library: v1.1.0

Address	Value (word)
0x1FF1FDB0	0xFF010100

RSS: v1.1.0

Address	Value (word)
0x1FF1FDAC	0xFF010100

Verification methods and expected values are summarized in section 3.1 of [4].

Secure initialization of the platform

The platform ensures its authenticity and integrity during the platform initialization. If the authenticity or integrity of the platform cannot be ensured, the platform goes into a locked state.

Conformance rationale:

Secure initialization is ensured considering that the STiROT is verifying the OEMuROT located in the external memory. Graphical view in Figure 2. Secure initialization is ensured, thanks to a multistep approach as the STiROT delegates the loading of the external memory to iLoader/ExtMemMgr.

iLoader and ExtMemMgr are installed in UserFlash in WRP (write protect) mode. OEMuROT is installed in an external memory.

After each reset, the TOE boots on the RSSs (Root Secure Services). RSSs are natively present in the product in the immutable area of the flash memory.

RSSs then jump into STiROT if the IROT_SELECT option byte selects STiROT (the PSA immutable Root of Trust of the platform).

The STiRoT then manages the secure boot of the application based on a 2-step approach:

- Step 1: STiROT launches the iLoader expecting the OEMuROT image to be loaded.
  If nothing is in the download slot, iLoader loads OEMuROT from the installation slot of the external memory to the SRAM, then launches a reset.
- Step 2: STiROT verifies the OEMuROT image loaded in Step 1 before launching it.
  - Verifies application integrity (before executing it) using the referenced SHA-256 value programmed in a secure flash area calculated at the end of the installation process.
  - Verifies application authenticity using ECDSA over curve ECC 256p1 using the public key stored in embedded flash OB keys. This authentication occurs each time the application is initially installed/ updated.

Residual information purging

The platform ensures that all SRAM and OB keys used by the platform, except SRAM not used by the platform, are erased using the method specified in section 5.2 of [5] before the memory is used by the platform or application again and before an attacker can access it.

Conformance rationale:

- The platform erases all its SRAM area before jumping to the application.
- The platform uses the HDP protection hardware feature to protect OB keys as described in section 5.4.4 of [5].
- The platform increments the HDP level before jumping to the application using the Security library (refer to section 4.2.2 of [4], in the *security library interface* subsection). Doing so, the platform clears access to the platform OB keys.
- The platform clears all its allocated SRAM by simply writing `0x0` on each allocated SRAM address.

3.3.2 Package “Security Services” Security Functional Requirements

Cryptographic KeyStore

The platform provides the application with a way to store secret keys such that even the application cannot compromise the authenticity and integrity of this data. This data can be used for cryptographic operations: encryption, decryption, authenticated encryption/decryption, signature, and verification.

Conformance rationale:

STiRoT supports Keystore. The platform uses Keystore to protect HDP-level keys (meaning OBKeys). OBKeys can be symmetric or asymmetric. For authenticity, STiROT uses ECDSA 256p1 over OBKeys at key installation within Keystore. For integrity, STiROT uses the SHA256 hash algorithm over OBKeys at each platform reset.

3.3.3 Package “Software Isolation” Security Functional Requirements

Software attacker resistance: Isolation of platform

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

Conformance rationale:

STiRoT enforces after each boot the platform HDP protection (refer to sections 5.5.4 and 5.5.5 of [5]) that establishes complete isolation between platform and application, preventing any application access to platform-sensitive assets (including STiROT code and data).

3.3.4 Additional Security Functional Requirements

Secure storage (internal storage)

The platform ensures that all data stored by the application, except for *data outside OBKeys HDPL1 and 2*, is protected to ensure its authenticity and integrity as specified in *FIPS PUB 186-4 for authenticity (ECDSA 256p1) and FIPS 180-4 for integrity (SHA256)*.

Conformance rationale:

Data authenticity is verified at data installation or update against the application public key (ECDSA 256p1 cryptographic algorithm). Integrity is checked via a SHA2-256 at each boot.

Secure encrypted storage (internal storage)

The platform ensures that all data stored by the application, except for *data outside OBKeys HDPL1 and 2*, is encrypted as specified in *NIST SP800-38A (AES CBC)* with a platform instance unique 256-bit key.

Conformance rationale:

STiRoT ensures that all data stored by the application within OBKeys is encrypted using an AES-CBC encryption policy with a 256-bit key based on AHK (keys generated on the first boot) of the product thanks to a TRNG.

Authenticity and integrity are covered by the “Secure Storage” SFR.

Secure installation of the application

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

Conformance rationale:

STiRoT detects any new application firmware version installation request and manages it in a secure way:

- Authenticity: ECDSA over ECC curve 256p1 against TOE user’s public key
- Decryption: AES CTR 128 bits against TOE user’s key

Secure update of the 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:

STiRoT detects any new application firmware version update request and manages it in a secure way:

- Authenticity: ECDSA over ECC curve 256p1 against TOE user’s public key.
- Decryption: AES CTR 128 bits against TOE user’s key.
- Version: STiROT uses an anti-rollback mechanism before granting the installation of a new application version.

3.4 SFRs for TOE_WITHOUT_STIROT configuration

3.4.1 Base PP Security Functional Requirements

Verification of Platform Identity

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

Conformance rationale:

The platform referred to in Section 1.2 provides the following unique identifications:

- Integrated circuit hardware revision DieID and DieID, at `0x5C001000` and `0x5C001002` addresses:

```
Die ID: 485, meaning STM32H7Sxx
```

Address	Value (halfword)
0x5C001000	0xX485

```
Revision ID: v1.2
```

Address	Value (halfword)
0x5C001002	0x1003

```
Product configuration: Crypto
(All crypto activated)
```

Address	Value (halfword)
0x5200281C	Bit 8 (SAES, CRYP, PKA available): 0b0 Bit 16,17,18 (MCE available): 0b000

- Immutable firmware secure versions:

```
Debug Authentication: v1.0.0
```

Address	Value (word)
0x1FF1FDC8	0xFF010000

```
Security Library: v1.1.0
```

Address	Value (word)
0x1FF1FDB0	0xFF010100

```
RSS: v1.1.0
```

Address	Value (word)
0x1FF1FDAC	0xFF010100

Verification methods and expected values are summarized in section 3.1 of [4].

Residual Information Purging

The platform ensures that user flash, SRAM, and OBKeys, except SRAM not used by the platform, are erased using the method specified in section 5.4.4 of [5] before the memory is used by the platform or application again and before an attacker can access it.

Conformance rationale:

- The Platform erases all its SRAM area before jumping to the application.
- The platform uses the HDP protection hardware feature to protect OBKeys. Refer to section 5.4.4 of [5].
- The platform increments the HDP level before jumping to the application using the Security library (refer to section 4.10.2 of [5]). Doing so, the platform clears access to the platform OB keys.
- The platform clears all its allocated SRAM by simply writing 0x0 on each allocated SRAM address.

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”, 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 3.2 to Section 3.4	All SFRs in this ST are taken from [1]. “Verification of Platform Identity” is included. “Secure Update of Platform” is not included (justification in ALC_FLR.2).
	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 Labelling 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 4.2	The platform evaluator determines whether the provided evidence is suitable to meet the requirement.
	ALC_FLR.2 Flaw reporting procedures	Section 3.1.1	The flaw reporting and remediation procedure is 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 A vulnerability analysis is performed by the platform evaluator 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. Penetration testing is performed by the platform evaluator assuming a potential attack of enhanced-basic.

4.2 PSA Security Function Mapping

Table 8. PSA Security Function Mapping

PSA security function	Covered by SESIP SFR	Supported by TOE_WITH_STIROT configuration	Supported by TOE_WITHOUT_STIROT configuration
F.INITIALIZATION	Secure initialization	Yes	No
F.SOFTWARE_ISOLATION	Software attacker resistance: Isolation of Platform	Yes	No
	Software attacker resistance: Isolation of application parts	No	No
F.SECURE_STORAGE	Secure encrypted storage	Yes	No
	Secure storage	Yes	No
	Secure external storage	No	No
F.FIRMWARE_UPDATE	Secure update of Platform	No	No
F.SECURE_STATE	Software attacker resistance: Isolation of Platform	Yes	No
	Secure initialization	Yes	No
	Secure update of Platform	No	No
F.CRYPTO	Cryptographic operation	Yes	Yes
	Cryptographic KeyStore	Yes	No
	Cryptographic random number	No	No
	Cryptographic key generation	No	No
F.ATTESTATION	Verification of Platform identity	Yes	Yes
	Verification of Platform instance identity	Yes	Yes
	Attestation of Platform Genuineness	No	No
	Attestation of Platform State	No	No
F.AUDIT	Audit log generation and storage	No	No
F.DEBUG	Secure debugging	Yes	Yes
F.PHYSICAL	Physical attacker resistance	Yes	Yes
Additional security functionality	Secure communication support	No	No
	Secure communication enforcement	No	No

5 Reference documents

Table 9. Reference documents

Reference	Definition
Evaluation documents	
[1]	*Security Evaluation Standard for IoT Platforms (SESIP), version 1.1 (June 2021), GlobalPlatform®, GP_FST_070*
[2]	*SESIP Protection Profile for Secure MCUs and MPUs, version 1.0 (Oct 2021), GlobalPlatform®, GPT_SPE_150*
[3]	*SESIP Profile for PSA Certified RoT Component Level 3, version 1.0 REL 02 (24/11/2022), Arm®, JSADEN018*
Development documents	
[4]	User manual *STM32H7Sxx security guidance for SESIP 3 Certification* (UM3290) revision 3
[5]	Reference manual *STM32H7Rx/Sx Arm®-based 32-bit MCUs* (RM0477) revision 6
[6]	*Authenticated Debug Access Control, version 1.0, Arm Limited, DEN0101*
[7]	*H7S ALC_CMC.1 and ALC_CMS.1 document v1.0 P5*
Standards	
[8]	*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 10. **Glossary**

Term	Definition
Application	Used in SESIP to refer to the components that are out of the scope of the evaluation.
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 defines a combination of the immutable platform Root of Trust and the updateable platform Root of Trust, considered the most trusted security component on the device.

7 Abbreviations

Table 11. **Abbreviations**

Term	Definition
RoT	Root of Trust
PSA	Platform security architecture
PSA-RoT	PSA Root of Trust

Revision history

Table 12. **Document revision history**

Date	Revision	Changes
15-Mar-2024	1	Initial release.
02-Apr-2024	2	Updated to align with security guidance revision 2.
14-Jun-2024	3	Updated to add some information on regression.

Contents

List of tables

List of figures