Technical note

STM32H533xx 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

This Security Target document describes the STM32H533 platform and the exact security properties of the platform that are evaluated against the GlobalPlatform® Security Evaluation Standard for IoT Platforms (SESIP) [1].

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

Table 1. **Protection Profile Reference and Conformance Claims for the 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 the TOE_WITHOUT_STIROT and TOE_WITH_STIROT configurations**

Reference	Value
Protection Profile name	SESIP Profile for PSA Certified RoT Component Level 3[7]
Protection Profile version	1.0 REL
Assurance claim	Refer to Section 3.1

1.1 Security Target Reference

This document: Technical note *STM32H533xx Security Target for Security Services* (TN1553), STMicroelectronics.

1.2 Platform Reference

Table 3. **Platform Reference**

Reference	Value
Platform name and version	Integrated circuit: STM32H533 (DieID= `0x478`), version 1.0 (RevID= `0x1000`) with full cryptographic configuration=(`0x40022428:bits 1;4;5=0`) Immutable firmware versions: • Configuration TOE_WITH_STIROT: – STiRoT version: v1.1.0 – Debug authentication version: v1.2.0 – Security library version: 1.0.0 • Configuration TOE_WITHOUT_STIROT: – Debug authentication version: v1.2.0 – Security library version: 1.0.0
Platform identification	STM32H533xx
Platform type	Microcontroller platform, with its 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**

Reference	Name	Version
UM3299[3]	*STM32H533xx security guidance for SESIP 3 Certification*	1
RM0481[4]	Reference manual *STM32H523/33xx, STM32H562/63xx, and STM32H573xx Arm®-based 32-bit MCUs*	2

1.4 Platform functional overview and description

The STM32H533 microcontroller is the SESIP-certified member of the STM32H5xx family of general-purpose microcontroller solutions (MCU). It ensures superior cyberprotection for cost- and power-conscious connected devices, as well as high-end core performance and peripheral integration.

The platform consists of an Arm® Cortex®-M33 based microcontroller with immutable firmware (also called Security Services) and with its peripherals, such as internal flash, protected flash, SRAM, crypto accelerators, and TRNG.

1.4.1 Platform security features and scope

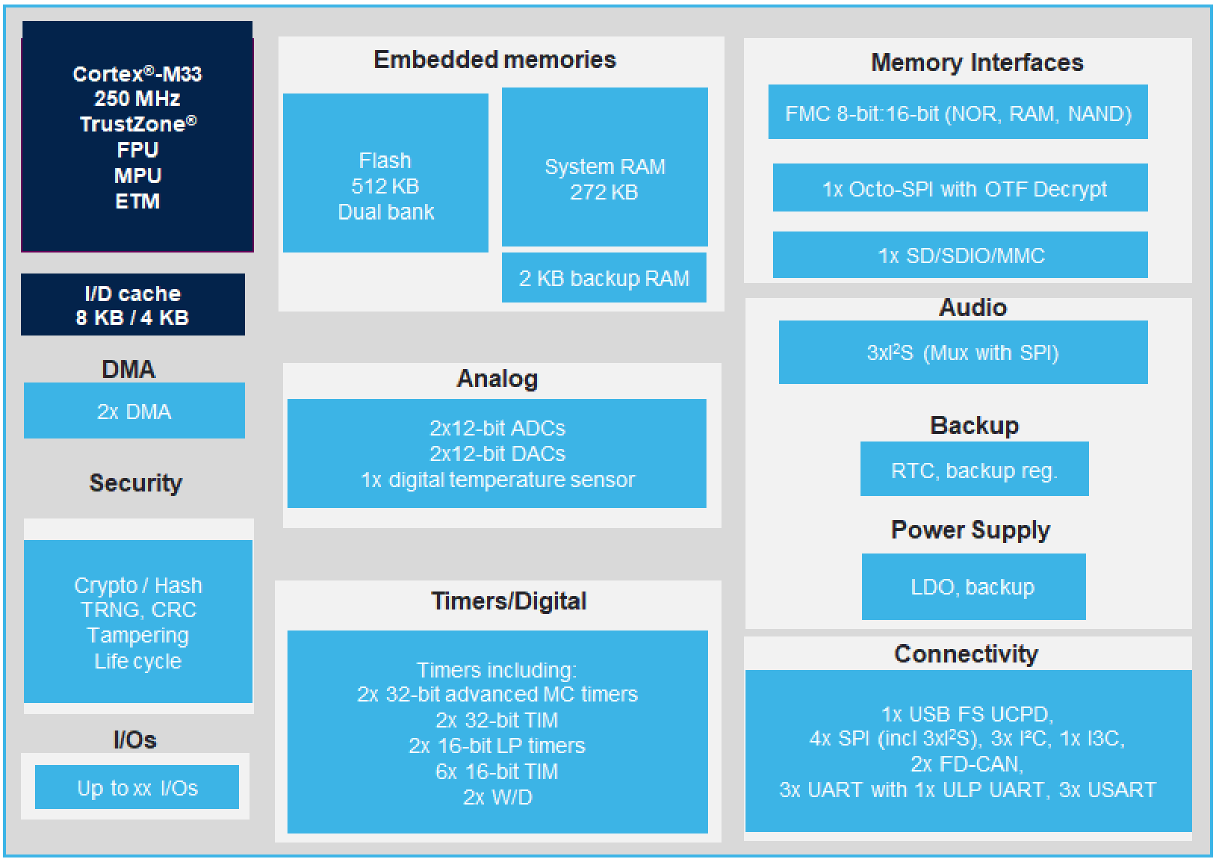
The STM32H533 microcontroller is designed with a comprehensive set of security features, some of them based on the standard Arm® TrustZone® technology. Those features include:

- 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)
- Security Services: Security Services are embedded in the system memory to manage the Root of Trust services. The immutable Root of Trust services handle 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 and TrustZone are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

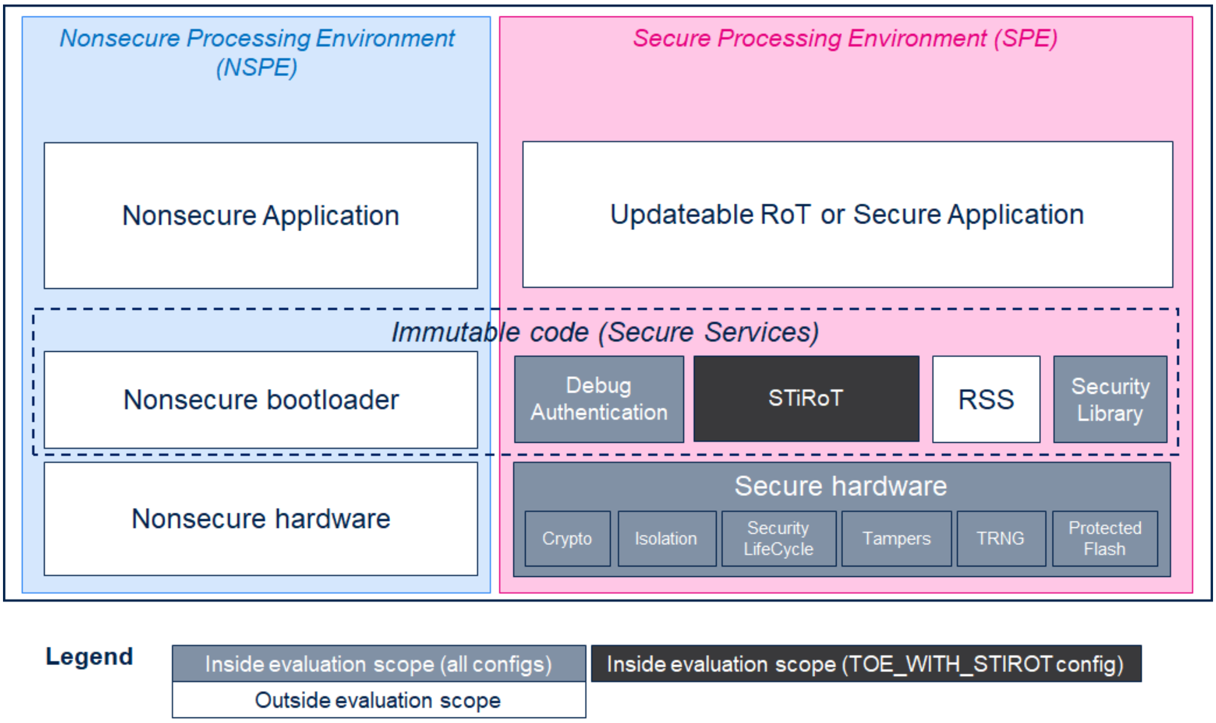
For more details, refer to Section 3 of [4]. An overview of the STM32H533 microcontroller is shown in the block diagram below.

Figure 1. STM32H533 block diagram



The figure below details the hardware and firmware used for the certification.

Figure 2. Detailed STM32H533 TOE scope


Nonsecure Processing Environment (NSPE)
Secure Processing Environment (SPE)
Nonsecure Application
Updateable RoT or Secure Application
Immutable code (Secure Services)
Nonsecure bootloader
Debug Authentication
STiRoT
RSS
Security Library
Secure hardware
Crypto
Isolation
Security LifeCycle
Tampers
TRNG
Protected Flash
Nonsecure hardware
Legend
Inside evaluation scope (all configs)
Inside evaluation scope (TOE_WITH_STIROT config)
Outside evaluation scope

The physical scope of the TOE is the STM32H533xx integrated circuit, identified as defined in Section 1.2. The hardware interfaces of the TOE are listed in Section 4.2 of [3].

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

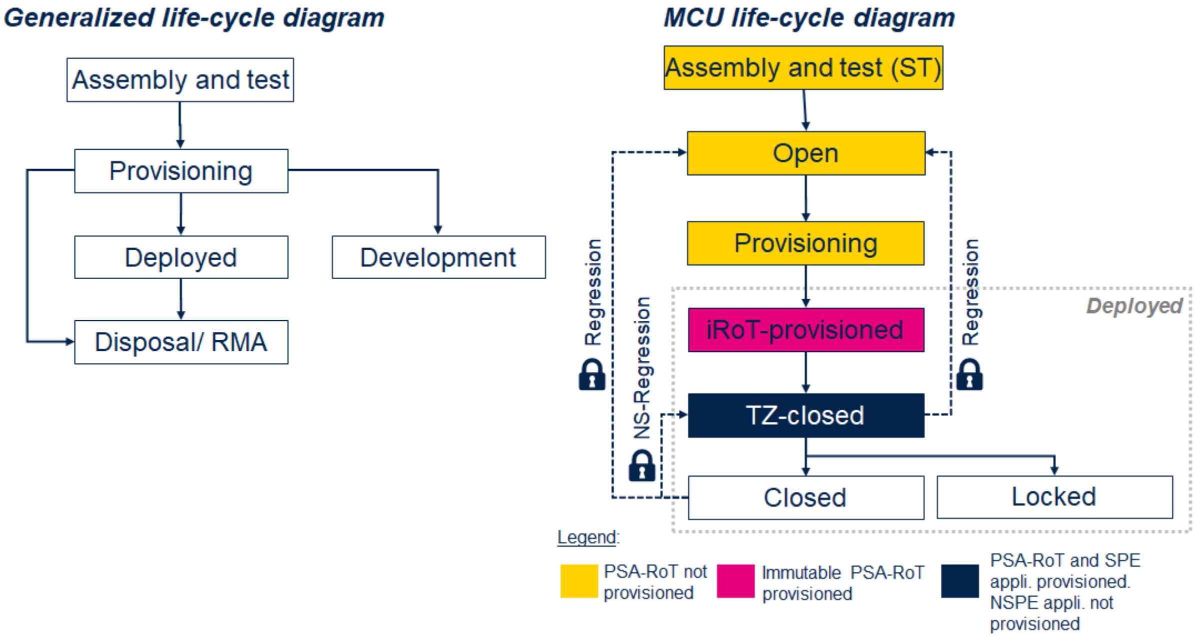
Component/ Interface	Description	Identification/ Version
STiRoT	The portion of immutable firmware that manages the secure boot and the secure firmware update of the application (code and/or related nonvolatile data) installed in the integrated user flash and option byte area	1.1.0
Security library	The portion of immutable firmware that manages the jump from iROT to BL or from iROT to the application	1.0.0
Debug authentication	The portion of immutable firmware in charge of secure debug reopening or secure regression (i.e. erasing memories content)	1.2.0
RSS	The portion of immutable firmware in charge of RSSe installation. RSS is not part of the TOE as neither STiRoT, debug authentication, nor security library run firmware from RSS. STiRoT and debug authentication set the full RSS section that owns RSS firmware as nonexecutable.	-
APIs	Refer to [4] to get the API descriptions.	2

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 3.11 of [4]. An overview of it can be found below.

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 at least set to *provisioned*, and the debug authentication service enabling the various regressions is always available unless the product state is *locked*.

For more details, refer to Section 3.2 of [3].

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 TOE_WITH_STIROT configuration, the 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 [3].
- The operating system or application code is expected to make use of the secure boot feature as described in Section 3.3 of [3].
- 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 [3] to meet the extra physical attacker resistance inherited objectives for the operational environment.

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

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) [1].

3.2 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 [8].

3.3 Common SFRs to every configuration

3.3.1 Base PP security functional requirements

Secure Debugging

The platform only provides debugger JTAG or SWD interface authenticated as specified in the PSA ADAC specification [5] with debug functionality.

The platform ensures that all data stored by the application, without exception, is made unavailable.

Debugging is only accessible after a Debug Authentication sequence.

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 [3].

When TOE is in life-cycle states *Provisioned*, *TZ-Closed*, or *Closed*, debug connection cannot be used. However, a trusted user can send to the platform via 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 certificate-verification debugging.

The OEM is responsible for defining Secure Debugging access rules thanks to device provisioning and generated/distributed certificates.

3.3.2 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 3.10.1 in [4].

Some of those algorithms are used by STiRoT.

Table 6. **TOE cryptographic operations**

Operations	Algorithm	Specification	Key lengths	Modes
Encryption, decryption	AES	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	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[1]	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[1]		
Cryptographic hash	SHA-2[2]	FIPS PUB 180-4	NA	SHA2-224, SHA2-256, SHA2-384, SHA2-512

1. *Standards for efficient cryptography: SEC1, SEC2*
2. *These algorithms must not be used when manipulating sensitive information.*

3.3.3 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 Cryptoengine 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.

3.3.4 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 platform identification and version mentioned in Section 3.4.1, the platform provides a *Unique Device ID* per chip. Refer to Section 59.1 of [4].

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 whether in nonvolatile memory (user flash) or volatile memory (SRAM). The regression service sets the platform product state to *OPEN*. That is the product state of the platform when it is delivered to the user. Refer to Section 4.2.2 in [3], *Debug Access* subsection, for details.

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

3.4 SFRs for TOE_WITH_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 revision ID, at `0x44024000` and `0x44024002` addresses:

  ```
  Die ID: 478, meaning STM32H533xx
  ```

Address	Value (halfword)
0x44024000	0xX478

```
Revision ID: v1.0
```

Address	Value (halfword)
0x44024002	0x1000

- Product configuration, halfword readable at the address `0x40022428` with bits set to values defined below:
- Bit 1: 0 (SAES available)
- Bit 4: 0 (AES available)
- Bit 5: 0 (PKA available)
- Immutable firmware versions:

```
STiRoT: v1.1.0
```

Address	Value (word)
0x0BF96084	0x01010000

```
Debug Authentication: v1.2.0
```

Address	Value (word)
0x0BF96060	0x01020000

```
Security Library: v1.0.0
```

Address	Value (word)
0x0BF9603C	0x01000000

Verification methods and expected values are summarized in Section 3.1 of [3].

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:

The Unique Boot Entry (UBE) option is configured (within the option bytes) to boot in system flash. After each reset the TOE boots on the STiRoT (the PSA immutable Root of Trust of the platform). The STiRoT then manages the secure boot of the application installed inside the user flash memory of the STM32H533 by:

- Verifying application integrity (before executing it) using the referenced SHA-256 value programmed in a secure flash area during the HDPL1 OB keys programming step
- Verifying application authenticity using ECDSA over curve ECC 256p1 using the public key stored in embedded flashOB Keys. This authentication occurs each time the application is installed.

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 7.6.12 of [4] 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 7.6.12 of [4].
- The platform increments the HDP level before jumping to the application using the Security library. Refer to Section 4.2.2 of [3] 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.

3.4.2 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.

Cryptographic Random Number Generation

The platform provides the application with a way based on *<list of entropy sources>* to generate random numbers as specified in [6] (SP 800-90B).

Conformance rationale:

The TOE includes an RNG peripheral compliant with NIST SP800-90B recommendations. The application must use this peripheral to generate true random numbers. Refer to RM0481[4] Section 32 True random number generator (RNG) for details.

3.4.3 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 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 Section 7.6.2 of [4] that establishes complete isolation between platform and application, preventing any application access to platform-sensitive assets (including STiRoT code and data).

3.4.4 Additional Security Functional Requirements

Secure storage (internal storage)

The platform ensures that all data stored by the application, except for *data outside OBKeys HDPL2*, 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)* with a platform instance unique 256-bit key.

Conformance rationale:

STiRoT provides a service to update securely application data in a dedicated embedded flash memory area known as OBKeys, which can be used by the application. 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. The application can update data by preloading a new encrypted blob in a download slot (data is updated during the next product reset).

Secure encrypted storage (internal storage)

The platform ensures that all data stored by the application, except for *data outside OBKeys HDPL2*, 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 derived from HUK within the flash OBKeys area.

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 securely:

- 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 securely:

- 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.

3.5 SFRs for TOE_WITHOUT_STIROT configuration

3.5.1 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 revision ID, at `0x44024000` and `0x44024002` addresses:

```
Die ID: 478, meaning STM32H533xx
```

Address	Value (halfword)
0x44024000	0xX478

```
Revision ID: v1.0
```

Address	Value (halfword)
0x44024002	0x1000

- Product configuration, halfword readable at the address `0x40022428` with bits set to values defined below:
- Bit 1: 0 (SAES available)
- Bit 4: 0 (AES available)
- Bit 5: 0 (PKA available)
- Immutable firmware secure versions:

```
Debug Authentication: v1.2.0
```

Address	Value (word)
0x0BF96060	0x01020000

```
Security Library: v1.0.0
```

Address	Value (word)
0x0BF9603C	0x01000000

Verification methods and expected values are summarized in Section 3.1 of [3].

Residual Information Purging

The platform ensures that user flash, SRAM, and OBKeys, without exception, are erased using the method specified in Section 7.6.11 of [4] before the memory is used by the platform or application again and before an attacker can access it.

Conformance rationale:

The Platform provides a service called regression supported by the debug authentication firmware stored in the TOE. This regression service erases all application code and data in embedded user flash, embedded volatile memory (SRAM), and OBKeys. Refer to Section 7.6.11 of [4] *Transition to Open state* subsection for details. Regression service is accessible via the platform debug port (SWD/JTAG).

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 3.3 to Section 3.5	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 will determine 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 will determine 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 will determine whether the provided evidence is suitable to meet the requirement.
	AGD_PRE.1 Preparative procedures	Section 1.3	The platform evaluator will determine 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 will determine whether the provided evidence is suitable to meet the requirement.
	ALC_CMS.1 TOE CM Coverage	Section 5	The platform evaluator will determine whether the provided evidence is suitable to meet the requirement.
	ALC_FLR.2 Flaw reporting procedures	Section 3.2	The flaw reporting and remediation procedure is described.
ATE: Tests	ATE_IND.1 Independent testing: conformance	Material provided to the evaluator	The platform evaluator will determine 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 7. 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 8. 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*
[7]	*SESIP Profile for PSA Certified RoT Component Level 3, version 1.0 REL (24/11/2022), Arm®, JSADEN018*
Development documents	
[3]	*STM32H533xx security guidance for SESIP 3 Certification (UM3299), revision 1*
[4]	*Reference manual STM32H523/33xx, STM32H562/63xx, and STM32H573xx Arm®-based 32-bit MCUs (RM0481), revision 2*
[5]	*Authenticated Debug Access Control, version 1.0, Arm Limited, DEN0101*
[8]	DM00882158, ST PRODUCT SECURITY INCIDENT RESPONSE TEAM (PSIRT) MANAGEMENT, revision 1.0
Standards	
[6]	*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 9. Glossary

Term	Definition
Application	Used in SESIP to refer to the components that are out of the scope of the evaluation.
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.
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.
Secure processing environment (SPE)	The processing environment that hosts the PSA-RoT, and any application RoT services.

7 Abbreviations

Table 10. Abbreviations

Term	Definition
NSPE	Nonsecure processing environment
PSA	Platform security architecture
PSA-RoT	PSA Root of Trust

Revision history

Table 11. Document revision history

Date	Revision	Changes
12-Dec-2024	1	Initial release.

Contents

1 Introduction 2
1.1 Security Target Reference 2
1.2 Platform Reference 2
1.3 Included guidance documents 3
1.4 Platform functional overview and description 3
1.4.1 Platform security features and scope 3
1.4.2 Life cycle 6
1.4.3 Use case 6
2 Security objectives for the operational environment 7
2.1 Platform objectives for the operational environment 7
3 Security requirements and implementation 8
3.1 Security assurance requirements 8
3.2 Flaw reporting procedure (ALC_FLR.2) 8
3.3 Common SFRs to every configuration 8
3.3.1 Base PP security functional requirements 8
3.3.2 Package “Security Services” security functional requirements 8
3.3.3 Package “hardware protections” security functional requirements 9
3.3.4 Additional security functional requirements 10
3.4 SFRs for TOE_WITH_STIROT configuration 10
3.4.1 Base PP Security Functional Requirements 10
3.4.2 Package “Security Services” Security Functional Requirements 11
3.4.3 Package “Software Isolation” Security Functional Requirements 12
3.4.4 Additional Security Functional Requirements 12
3.5 SFRs for TOE_WITHOUT_STIROT configuration 13
3.5.1 Base PP Security Functional Requirements 13
4 Mapping and Sufficiency Rationales 14
4.1 SESIP3 Sufficiency 14
4.2 PSA Security Function Mapping 15
5 Reference documents 16
6 Glossary 17
7 Abbreviations 18
Revision history 19
List of tables 21
List of figures 22

List of tables

Table 1. Protection Profile Reference and Conformance Claims for the TOE_WITH_STIROT configuration 2
Table 2. Protection Profile Reference and Conformance Claims for the TOE_WITHOUT_STIROT and TOE_WITH_STIROT configurations 2
Table 3. Platform Reference 2
Table 4. Guidance documents 3
Table 5. Software components and interfaces of the TOE 5
Table 6. TOE cryptographic operations 9
Table 7. PSA Security Function Mapping 15
Table 8. Reference documents 16
Table 9. Glossary 17
Table 10. Abbreviations 18
Table 11. Document revision history 19

List of figures

Figure 1. STM32H533 block diagram 4
Figure 2. Detailed STM32H533 TOE scope 5
Figure 3. Connected platform life cycle overview 6