STM32MP2xxC/xxF security target

Document information

This document provides information required for evaluating security properties of the STM32MP2xxC/xxF devices in compliance with GlobalPlatform *Security Evaluation Standard for IoT Platforms (SESIP)*, version 1.2.

1 Introduction

This document defines the scope of the target of evaluation (TOE) also called *Platform*, the applicable security evaluation standard for IoT platforms (SESIP) profile, and the *SESIP*-related assurance claim. It describes the Platform (target of evaluation) with respect to the security assurance requirements (SARs) and security functional requirements (SFRs) of the applicable protection profile.

The following tables provide information on the applicable SESIP profile (SP) and conformance claims.

Table 1. SESIP profile 1

Reference	Value
SP name	*SESIP Profile for Secure MCUs and MPUs*, [SP1]
SP version	1.1
Package claim	Base SP, Security services, Software isolation, and Hardware protections
Assurance claim	Refer to Section 3.1: Security assurance requirements.

Table 2. SESIP profile 2

Reference	Value
SP name	*SESIP Profile for PSA Certified RoT Component Level 3*, [SP2]
SP version	1.0
Optional and additional *SFR*s	Base profile
Assurance claim	Refer to Section 3.1: Security assurance requirements.

The STM32MP2xxC/xxF microprocessor unit (MPU) devices are second-generation microprocessors enabling security for Industry 4.0. They are based on Arm® Cortex®-A35 and Cortex®-M33 cores. The Cortex®-A35 core executes the embedded ROM code in isolation thanks to the Arm® TrustZone® architecture extension and the STM32 *Compartment ID* technology. The devices also include embedded SRAM, nonvolatile fuses, cryptographic accelerators, and a true random number generator.

The *xx* wildcards in the generic STM32MP2xxC/xxF product number denote its different versions. The first *x* is a generic substitute of 1, 3, or 5. The second *x* is a generic substitute of 1, 3, 5, or 7. The devices with their specific part numbers differ in their integrated peripherals such as video and display interfaces, serial communication interfaces, GPU, NPU, in the number of cores, and so on. The *C* and *F* pertain to the core maximum cadence. For details, refer to documents listed in Table 4. Guidance documents.

arm

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

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

The Arm word and logo are trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere. All rights reserved.

1.1 Security target reference

This document: *STM32MP2xxC/xxF security target* (ST0057) revision 3, STMicroelectronics.

1.2 Platform references

The following table provides *Platform* references.

Table 3. Platform references

Reference		Value
Platform name		STM32MP2xxC/xxF Arm®-based 32/64-bit MPUs
Platform version	STM32MP21xC/xF	1.1
	STM32MP23xC/xF	2.2
	STM32MP25xC/xF	2.2
Platform identification	STM32MP21xC/xF	ID: 0x503
	STM32MP23xC/xF	ID: 0x505, RPN: bit[1] = 1
	STM32MP25xC/xF	ID: 0x505, RPN: bit[1] = 0
Platform type		General-purpose microprocessor device for IoT, industrial, or consumer applications. It focuses on platforms with security functionalities. Refer to the documents listed in Table 4. Guidance documents.

1.3 Included guidance documents

The following table lists the documents included with *Platform*.

Table 4. Guidance documents

Category	Name and revision	Reference
Security guidance	SG0052, *STM32MP2xxC/xxF security guidance for SESIP level 3 certification*, revision 2	[SG]
Device reference manuals	RM0506, *STM32MP21xx advanced Arm®-based 32/64-bit MPUs*, revision 1 RM0457, *STM32MP23xx/25xx advanced Arm®-based 32/64-bit MPUs*, revision 5	[RM]
Device datasheets	DS14556, *Arm® based Cortex®-A35 1.5 GHz + Cortex®-M33 MPU with TFT, 2× USB 2.0, crypto*, revision 1, (datasheet of STM32MP21xC/xF) DS14634, *Arm® based dual Cortex®-A35 1.5 GHz + Cortex®-M33 MPU, AI, 3D GPU, video decoder, TFT/DSI/LVDS, USB 2.0, crypto*, revision 3, (datasheet of STM32MP23xC/xF) DS14284, *Arm® based dual Cortex®-A35 1.5 GHz + Cortex®-M33 MPU with AI, 3D GPU, video encoder/decoder, TFT/DSI/LVDS, USB 3.0, PCIe®, crypto*, revision 5, (datasheet of STM32MP25xC/xF)	[DS]
Device errata sheets	ES0628, *STM32MP211x/3x/5x device errata*, revision 1 ES0598, *STM32MP25x/STM32MP23x device errata*, revision 5	[ES]

1.4 Platform component functional overview and description

This section defines the *TOE* as a subset of hardware (physical scope) and software (logical scope) components of the STM32MP2xxC/xxF devices. It describes security features and their usage, life cycle, and a use case.

The *Platform* provides the necessary hardware and immutable code building blocks for the *Platform* integrator to build a connected platform as defined in [SP1]. It can also be a basis for further composition of evaluation activities.

1.4.1 Platform security features and scope

The *Platform* as defined in Section 1.4.1.1 and Section 1.4.1.2 supports the following security features, some of which based on standard Arm® TrustZone® technology:

- Resource isolation using privilege mode and Armv8.1-M main line security extension of the Arm® Cortex®-A35 core, with securable I/Os, memories, and peripherals
- Following any device reset, the *Platform* executes its embedded secure ROM code, ensuring the second stage of the microprocessor trusted boot chain. For modes of operation, refer to Section 3.2.1.2: Secure initialization of Platform.
- Residual information purging for life cycle handling.

Optional packages in [SP1] are selected to accommodate the *Platform* context of use:

- Cryptographic operations, based on hardware cryptographic accelerators (SAES, PKA).
- True random number generator (RNG) that provides full entropy outputs to the application.
- Hardware protection to handle hostile environments.
- Isolation mechanisms controlling accesses between certified and noncertified parts of the device software.

Further considerations regarding the security features:

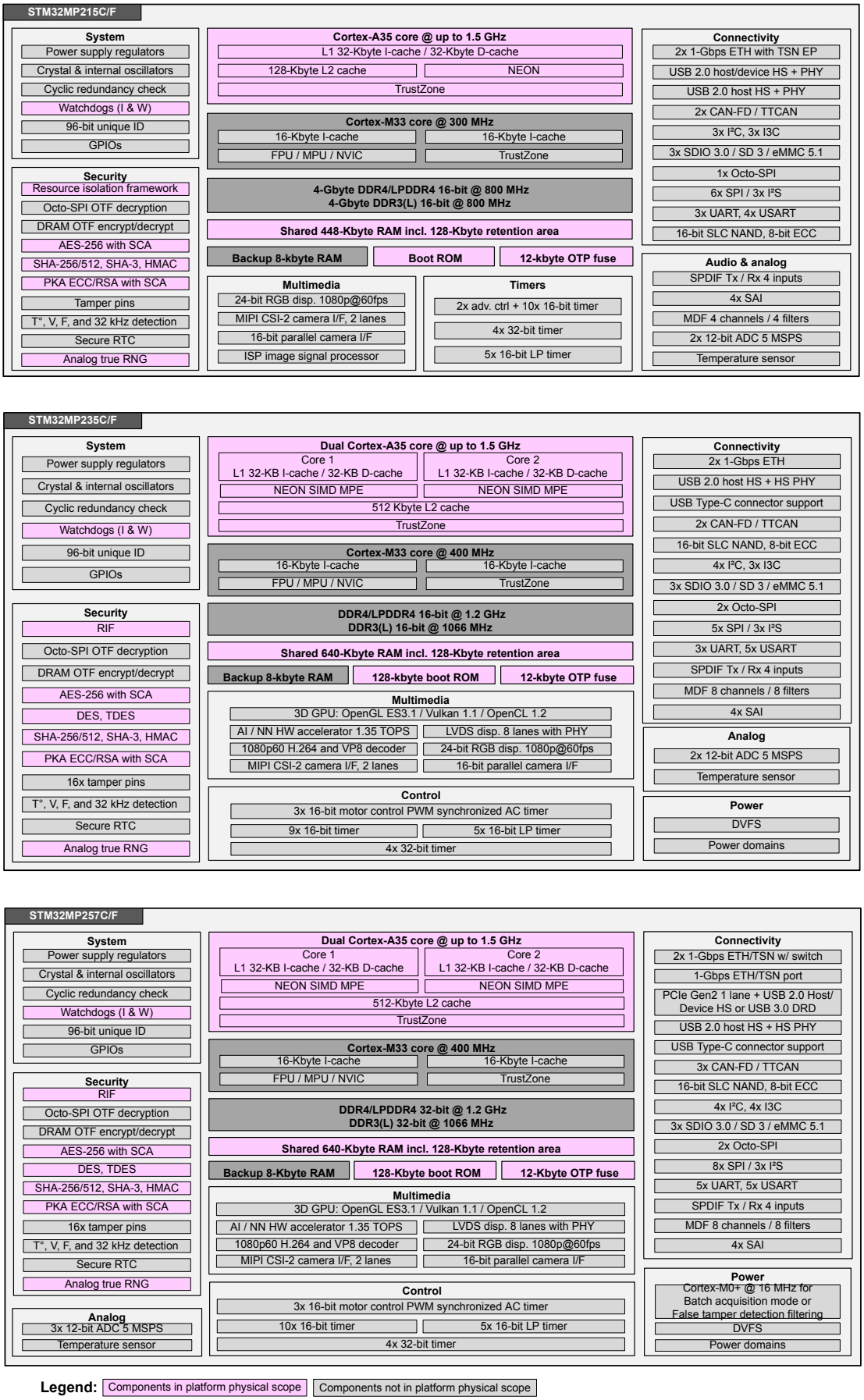
- Secure debugging of the *Platform* is implemented but not accessible to the users. Debugging of the non-*TOE* application can be activated using an authenticated boot image in certified configuration (no debug by default).
- Secure update of the *Platform* is not applicable since the *Platform* does not support the patching of the embedded ROM.
- The Cortex-M33 core executing in *SPE* is outside the *Platform* logical scope (see Section 1.4.1.2: Platform logical scope).

For the *Platform* usage, refer to [SG].

1.4.1.1 Platform physical scope

The *Platform* physical scope of is a subset of product hardware functional blocks as indicated in the following figure.

Figure 1. **Platform physical scope**



The following table summarizes the *Platform* hardware components. They are documented in [RM] and [ES].

Table 5. Hardware components and interfaces of *TOE*

Component / Interface	Description(1)	Identification
CPU	Cortex-A35 subsystem (with MMU and caches)	Hardware revision.(2)
Communication ports	-	
Memories	ROM, SYSRAM, RETRAM(3)	
Infrastructure	RCC, PWR, BSEC, RIFSC, RISAB1/2/5, IWDG1/2, TAMP, SYSCFG, STGEN, FMC, GPIO	
Cryptography	**STM32MP21xC/xF:** SAES, PKA, CRYP1/2, HASH1/2, RNG1/2 **STM32MP23xC/xF, STM32MP25xC/xF:** SAES, PKA, CRYP1/2, HASH, RNG	

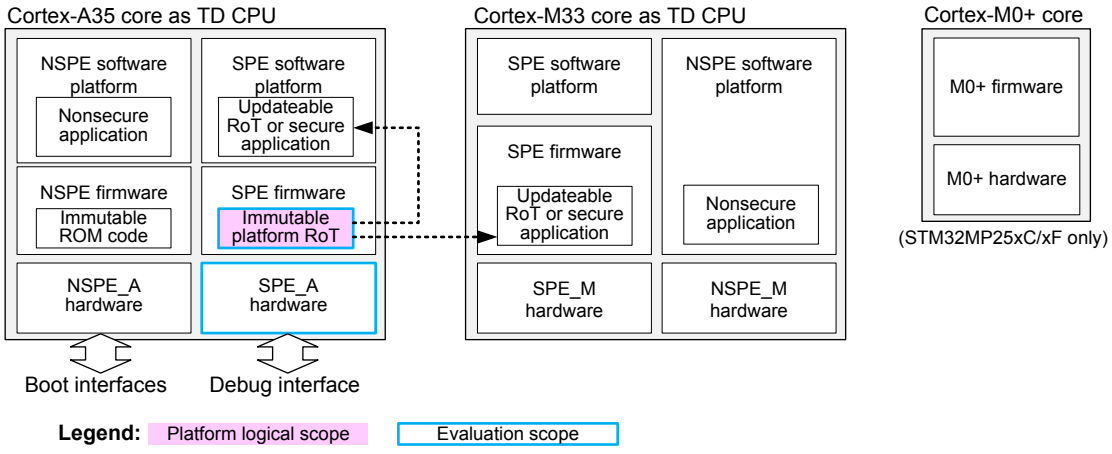
1. *Refer to [RM] for different functional blocks listed in this column.*
2. *Refer to the Platform version in Table 3. Platform references.*
3. *SYSRAM and RETRAM are specific SRAM assets.*

1.4.1.2 *Platform logical scope*

The logical scope (software components) of the *Platform* covers software components executed in the secure processing environment (SPE) and controlling or using the hardware components within the *Platform* physical scope presented in Section 1.4.1.1: Platform physical scope.

The following figure represents the platform logical scope.

Figure 2. Platform logical scope



The following table summarizes the software components within the *Platform* logical scope.

Table 6. Software components and interfaces of the *TOE*

Component/interface	Description	Identification/Version
Secure boot ROM code	Immutable root of trust (RoT) with boot code run on Cortex-A35 core in *SPE*, root parameters, managing, enforcing, or monitoring isolation hardware resources related to the immutable *RoT* functionality. It comprises code for controlling cryptographic operations involving random number generation, and code for security life cycle functions summarized in Section 1.4.1: Platform security features and scope.	Same as silicon(1)

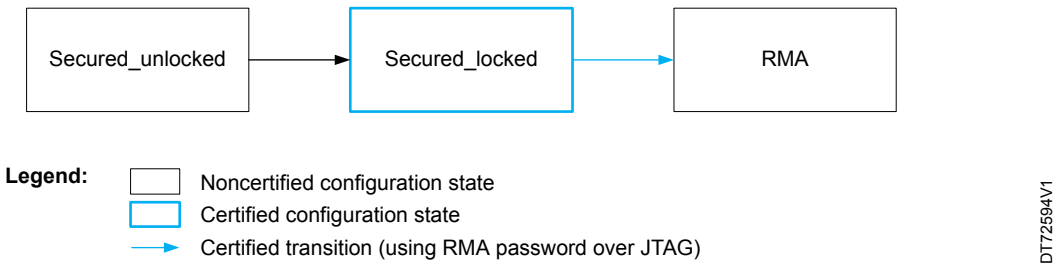
1. *Refer to Platform version in Table 3. Platform references.*

No additional nonplatform hardware, software, or firmware is required to support the security claims described in this document.

1.4.2 Life cycle

The product life cycle presented in the following figure is used in the *SFR*s when references to a life cycle are required. The on-the-field product must be configured in **Secured_locked** state, in accordance with [SG], section *Secure installation*.

Figure 3. Reference product life cycle



STMicroelectronics delivers **Secured_unlocked** STM32MP2xxC/xxF devices without *RoT* keys or RMA password provisioned. The provisioning of those is described in [SG], section *Software setup procedure*.

The integrity and confidentiality of the *Platform* during the life of the STM32MP2xxC/xxF devices are ensured because the ROM part of the *TOE* and the *TOE* secrets are hidden from the integrator code or debugger.

The transition to **RMA** state is initiated via the JTAG interface by the integrator using their own 128-bit password. After the **X** consecutive wrong password attempts (X < 5) by the integrator, the RMA sequence always fails.

As stated in [SG], section *User-accessible functions and privileges (AGD_OPE.1.1C)*, subsection *Nonvolatile storage of secrets in OTP*, the integrator must store their security-sensitive information (secret AES keys, private ECC or RSA keys) in OTP words 256 to 367, to prevent secret information from leaking when the device transits to **RMA** state.

1.4.3 Use case

The *Platform* is intended for integrators who wish to implement a secure boot with the necessary *RoT* services.

The *Platform* in certified configuration can securely be used under the following ***Any user*** and ***Any code*** environmental conditions:

- **Any user**: The device may physically be accessed by an unknown or untrusted user, in an environment where access to the device cannot be sufficiently controlled or even in a more hostile environment.
- **Any code**: The device may execute code that is unknown to the *Integrator*.

1.4.4 Required non-Platform assets

The device ROM code is executed in different use cases that may require specific configuration managed by a trusted updateable *RoT* code outside the *Platform* logical scope. The Cortex-A35 core immutable ROM code executing in *SPE*, as part of the *Platform* logical scope, points to this updateable *RoT* code software component, as presented in Figure 2. Platform logical scope.

Consequently, the trusted firmware that is outside the *Platform* logical scope must ensure at least the following steps:

- Configure the resource isolation framework (RIF).
- Keep the jump address to DStandby wake-up image unchanged (see Section 3.2.1.2: Secure initialization of Platform for details).

The required non-*Platform* trusted firmware expectations (ASE_INT.1.6C) are exhaustively described in [SG], section *Available interfaces and methods of use (AGD_OPE.1.2C and AGD_OPE.1.3C)*.

2 Security objectives for the environment

For the *Platform* to fulfill its security requirements, the operational environment (technical or procedural) must fulfill the following objectives:

Table 7. Security objectives for the environment

Security objective ID	Security objective description
Secure acceptance	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: Platform references.
Security guidance application	The development and operational environment shall implement protections of any material it handles and involved in the *Platform* security (such as firmware authentication key) at the same level of security as the *Platform*.
Secure boot usage	The operating system or application code is expected to use the secure boot feature as described in Section 3.2.1.2: Secure initialization of Platform.
Sensitive material secure handling	The *Platform* integrator must follow all *Platform* security guidelines provided in the *Platform* documentation or provide a justification and a workaround ensuring an equivalent level of security.

2.1 Inherited objectives for the operational environment

No inherited objectives are applicable as the *Platform* is not composite.

3 Security requirements and implementation

This section claims the compliance of the *Platform* with specific *SAR*s and *SFR*s described in [SESIP] and [SP1], and describes their implementation.

3.1 Security assurance requirements

This document claims compliance with the security assurance requirements defined in [SESIP], section 4 *Security assurance requirements*, item *SESIP Assurance Level 3 (SESIP3)*.

3.1.1 Flaw reporting procedure (ALC_FLR.2)

The *SFR* entitled *Secure update of platform* is not applicable as the *Platform* does not support the patching of the immutable ROM firmware. Outside the *Platform*, the integrator can implement their own secure update mechanism in their code, leveraging the boot image revocation method (see [SG] section 4.2.1 for details).

In accordance with the requirement ALC_FLR.2 for a flaw reporting procedure including a process to generate and distribute any required update, the *Developer* defines the procedure in [PSIRT].

3.2 Security functional requirements (SFRs)

This section claims compliance with respect to the security functional requirements defined in [SP1], section 4.2 to 4.8.

3.2.1 *Base SP* package of SFRs

This section describes how the *Platform* fulfils the *Base SP* package of *SFR*s.

3.2.1.1 Verification of Platform identity

The *Platform* is identified based on the unique device identifiers. These include their part number and their revision.

Conformance rationale

The following table lists the identifiers of the *Platform* referred to in Section 1.2: Platform references.

Table 8. Platform identifiers

Bitfield[logical bit range] / [physical bit range]	Register / Address	Bitfield value	Comment
STM32MP21x			
DEV_ID[11:0] / [11:0]	SYSCFG_IDC / 0x5423 6400	0x0503	STM32MP21xx, major revision 1
REV_ID[5:0] / [5:0]	ID / OTP 102	0x9	Major revision 1, minor revision 1
STM32MP23x			
DEV_ID[11:0] / [11:0]	SYSCFG_IDC/0x5423 6400	0x0505	STM32MP23x/25xx, major revision 2
REV_ID[5:0] / [5:0]	ID / OTP 102	0x12	Major revision 2, minor revision 2
RPN[0] / [1]	RPN_CODING / OTP 9	1	STM32MP23xx
STM32MP25x			
DEV_ID[11:0] / [11:0]	SYSCFG_IDC / 0x5423 6400	0x0505	STM32MP23x/25xx, major revision 2
REV_ID[5:0] / [5:0]	ID / OTP 102	0x12	Major revision 2, minor revision 2
RPN[0] / [1]	RPN_CODING / OTP 9	0	STM32MP25xx

Note: A method how to read OTP addresses 9 and 12 is described in [SG], section 3.1, subsection Secure acceptance.

3.2.1.2 *Secure initialization of Platform*

The *Platform* ensures its integrity and authenticity during the *Platform* initialization. If the *Platform* integrity and authenticity cannot be ensured, the *Platform* transits to *locked* state.

Conformance rationale

Following any device reset, the *Platform* executes its embedded secure ROM code, ensuring the second stage of the microprocessor trusted boot chain. There are different options of how to securely initialize the *Platform*. Refer to the following table and figure for details.

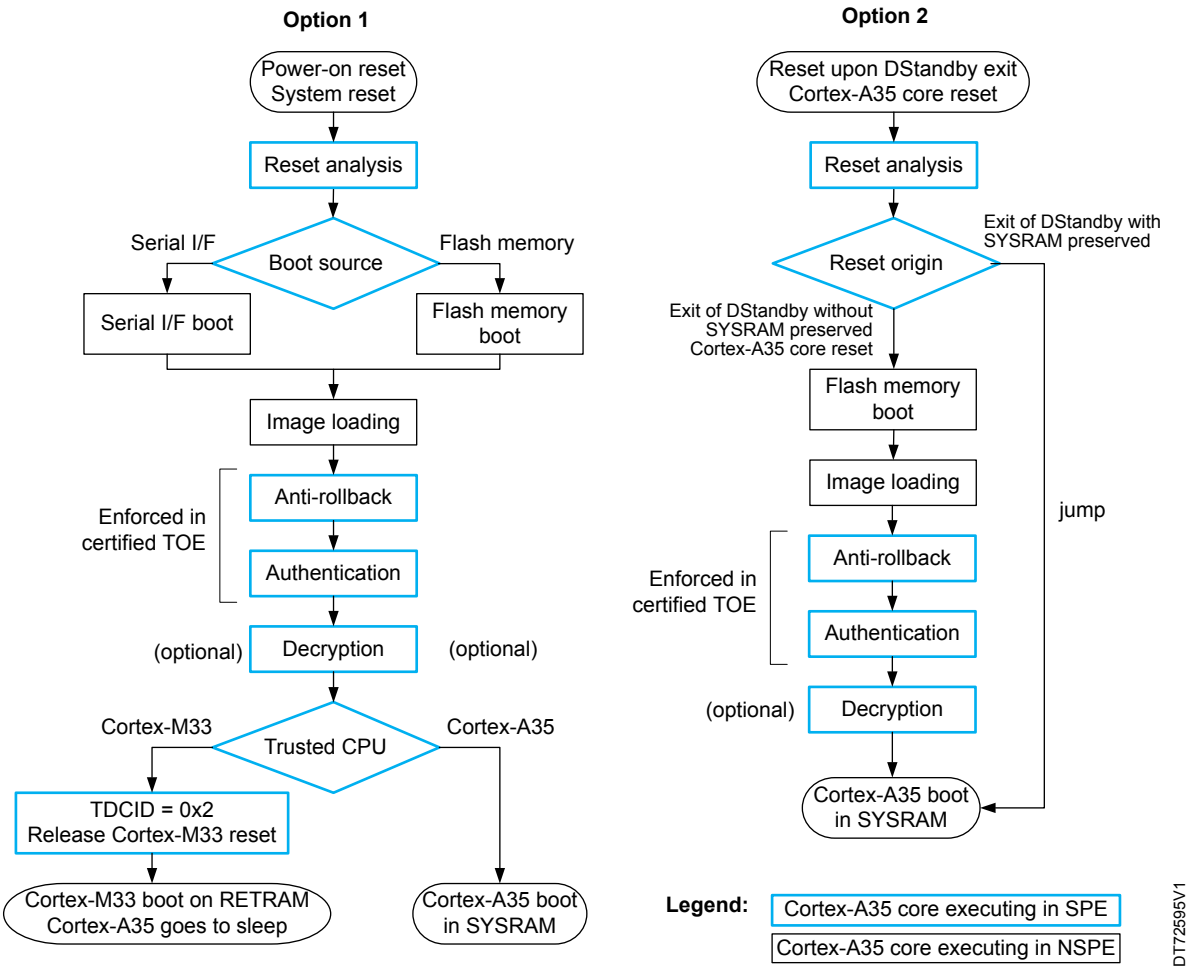
Table 9. Platform secure initialization

Option	Device reset	SYSRAM preserved	Jump to	Cortex-A35 core executes alone	Comment
1	Power-on reset or System reset	No	cold boot image (Cortex-A35)	Yes	Cortex-A35 core is the *TD CPU*. The cold boot image comes from the flash memory or serial interface.
			flash boot image (Cortex-M33)		Cortex-A35 core makes the Cortex-M33 core the *TD CPU*. The boot image only comes from the flash memory interface.
2	Cortex-A35 core reset	No	flash boot image (Cortex-A35)	Yes	-
	Reset upon DStandby exit	No		No	The isolation methods are described in Section 3.2.3.
		Yes	DStandby wake-up image		

The *Platform* authenticity and integrity are enforced by respecting the following principles:

- **For jumps to cold boot or flash boot image:** The *Platform* transfers the control to a secure bootloader code verified for integrity and authenticity using elliptic-curve digital signature. The bootloader code (SHA384 on STM32MP21, otherwise SHA-256) digest is always signed using an active ECDSA private key (384-bit on STM32MP21, otherwise 256-bit), based on NIST curves (prime384v1 or brainpoolP384t1 on STM32MP21, otherwise prime256v1 or brainpoolP256t1). The *Platform* can choose one among eight ECDSA public keys when verifying the bootloader code signature. The selected key is provided as cleartext in the boot image header, with its SHA-256 digest protected in the *Platform* on-chip fuses.
- **For the jump to DStandby wake-up image:** When the trusted application transfers the control to the firmware that executes on exiting DStandby, this firmware can not be modified by any code or any user while the device is in low-power mode (SYSRAM in read-only retention mode).

Figure 4. Platform secure initialization diagram



The execution steps in NSPE such as the image loading (in ROM) are outside the *Platform* logical scope (see Section 1.4.1.2). The transition after the *Image loading* block is the TOE *Boot image interface* as described in [SG], section 4.2.2 *Available interfaces and methods of use (AGD_OPE.1.2C and AGD_OPE.1.3C)*

.

Anytime during the *Platform* ROM execution, any fuse disturbance prevents the boot chain execution, forcing an application reset. In this way, any threat to impair the *Platform* authenticity or integrity leads to a lock state in which the application secrets stored in the on-chip fuses are inaccessible, and in which the debugger and test modes are disabled until the application reset. Refer to [RM], section *BSEC functional description*.

3.2.1.3 ~~***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.~~

Non-conformance rationale:

The absence of this functionality is justified in Section 3.1.1: Flaw reporting procedure (ALC_FLR.2).

3.2.1.4 *Residual information purging*

After using any secrets, the *Platform* purges or locks the corresponding memory locations before an attacker can access them.

Conformance rationale

The measures for preventing an attacker from accessing residual information used by the *Platform* are:

- The device reset erases the content of the specific SRAM assets listed in Table 5. Hardware components and interfaces of *TOE* before the *Platform* can use them.
- After the use of SRAM and register locations that contain secrets, the *Platform* replaces their contents with random values.
- Before jumping to the authenticated application, the *Platform*:
  - Clears all allocated embedded SRAM locations by writing zeros to each address
  - Applies a sticky-reload lock to its secrets stored in BSEC

3.2.1.5 ~~*Secure debugging*~~

~~The *Platform* only provides <list of endpoints> authenticated as specified in <specification> with debug functionality.~~

~~The *Platform* ensures that all data stored by the application, with the exception of <exceptions>, is made unavailable.~~

Non-conformance rationale

This *SFR* is removed because the secure Cortex-A35 debug feature, though implemented, is not accessible to users while the *Platform* is executing.

Regarding other debug features of the device, the *Platform* is certified with all debug features disabled after a cold boot reset (see Section 1.4.2: Life cycle). This debug protection can be permanently enabled until the next reset by setting the DENLOCK bit in the BSEC_LOCKR register of the BSEC peripheral. The *Platform* automatically sets this bit when the debug_lock fuses are programmed in the OTP word 18 (see [SG], section *User-accessible functions and privileges (AGD_OPE.1.1C)*, subsection *Secure debug*).

If the user activates any debug capability on the device following a cold boot as defined in Section 3.2.1.2: Secure initialization of Platform, debugging of the *Platform* ROM code remains inaccessible.

3.2.2 *Security services* package of SFRs

This section describes how the *Platform* fulfils the requirements of the *security services* package of *SFR*s.

3.2.2.1 *Cryptographic operation*

The *Platform* supports the cryptographic operations listed in the following table, with details on used algorithms, key lengths, and modes..

Conformance rationale

When the *Platform* is not executing, the application can use the SAES and PKA peripherals to access cryptographic operation resources. For more details, refer to [RM], sections *Secure AES coprocessor (SAES)* and *Public key accelerator (PKA)*.

The devices also provide cryptographic operations without side-channel attack resistance, listed in the following table. These cryptographic operations must not be used to process data within the logical scope of the *Platform* (subject to the *Platform* SESIP assurance level 3 certification). For more details, refer to [RM], section *Hash processor (HASH)*.

Table 10. Cryptographic operations outside the Platform

Operations	Algorithm	Specification	Key lengths	Modes
Cryptographic hash	SHA-2	FIPS PUB 180-4	N/A	SHA-256, SHA-384, SHA-512
	SHA-3	FIPS PUB 202	N/A	SHA3-256, SHA3-384, SHA3-512, SHAKE128, SHAKE256

Table 11. Platform cryptographic operations

Operations	Algorithms	Specifications	Key lengths in bits	Modes
Encryption, decryption	AES(1)	FIPS PUB 197 NIST SP800-38A	128, 192(2), 256	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(3)	RSA(4)	IETF RFC 8017 NIST SP800-56B FIPS PUB 186-4	Up to 4096	RSA 2048, 3072, 4096
Signature	ECDSA	ANSI X9.62 IETF RFC 7027 FIPS PUB 186-4 SEC 1, SEC 2(5)	Up to 640	**Nist:** P256, P384, P521 **Brainpool:** bp256r1, bp384r1, bp512r1 **SEC 2**: secp256k1, secp256r1, secp384r1, secp521r1
ECC scalar multiplication(6)	ECDH ECIES	ANSI X9.42 ANSI X9.63 FIPS PUB 186-4 SEC 1, SEC 2(5)		

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. *Only available on STM32MP21xx devices.*
3. *Signature, decryption, and key agreement.*
4. *Other operations not written in this table (like RSA CRT exponentiation or ECDSA signature verification) are not protected against side channel attacks.*
5. *SEC1 and SEC2 are standards for efficient cryptography.*
6. *Such as public key generation, key agreement, shared secret generation.*

3.2.2.2 *Cryptographic key store*

The *Platform* provides a secure method to store cryptographic keys, ensuring that even the application cannot compromise their confidentiality. These keys can be used for cryptographic operations involving the AES algorithm, as listed in Table 11. Platform cryptographic operations.

Conformance rationale

The *Platform* provides hardware mechanisms to protect the confidentiality of AES keys of sizes as defined in Table 11. Platform cryptographic operations. When the user encrypts the AES keys in the SAES peripheral using the derived hardware unique key (DHUK), they can only be decrypted on this specific device. The decrypted keys are accessible solely within the SAES write-only key registers. If the application attempts to overwrite any part of a key stored across multiple registers, all registers containing that key are automatically erased.

The *DHUK* is never exposed to application code or debuggers and is exclusively usable within the side-channel protected SAES peripheral. For more information, refer to [RM], section *SAES operation with wrapped keys*.

3.2.2.3 *Cryptographic random number generation*

The *Platform* provides the application with a method based on an analog live entropy source to generate random numbers, as specified in [NIST].

Conformance rationale

The *Platform* includes an RNG peripheral compliant with the recommendations of [NIST]. When the *Platform* is not executing then to generate true random numbers, the application must use this peripheral. Refer to [RM], section *True random number generator (RNG)* for details.

3.2.3 *Software isolation* package of SFRs

This section describes how the *Platform* fulfils the requirements of the *Software isolation* package of *SFR*s.

3.2.3.1 Software attacker resistance: isolation of Platform

The *Platform* provides isolation between the application and the platform itself, so that an attacker who is able to run code as an application on the platform cannot compromise the other security functional requirements.

Conformance rationale

The *Platform* employs the following isolation measures to ensure that the application cannot compromise the secure initialization of the *Platform* as defined in Section 3.2.1.2: Secure initialization of Platform:

- Before setting the D1 domain to DStandby mode, the application must configure the RIF as specified in [SG], section 4.2.1, subsection *Resource Isolation*. Failure to do so causes the *Platform* ROM code to enter a locked state. Trusted firmware can handle this configuration. This mandatory RIF configuration guarantees that no active master accesses the same resources as the *Platform* ROM code and ensures exclusive access to its peripherals.
- After a cold boot, the *Platform* ROM code and certain nonvolatile assets in the BSEC peripheral are hidden from any application code. See [RM], section *BSEC local access control* for details.
- The *Platform* ROM code running in the secure processing environment (SPE) manages the RIF configuration to protect itself from ROM code running in the nonsecure processing environment (NSPE), as illustrated in Section 3.2.1.2: Secure initialization of Platform.
- As defined in [SG], section 4.2.1, subsection *Jump to DStandby wakeup image*, the *Platform* verifies that backup register 11 is secure and contains a valid jump address pointing to the retention code in the SYSRAM area. The required non-*Platform* trusted firmware (see Section 1.4.4: Required non-Platform assets) must not modify the value written by the Cortex®-A35 core in backup register 11 when executing secure firmware.
- In all operational modes, the *Platform* ROM code manages the configuration of the RNG peripheral.
- According to [RM] section *RISAB implementation*, only the Cortex®-A35 core can program the firewalls protecting SYSRAM.

These protections apply regardless of whether the application selects the secure Cortex®-A35 core or secure Cortex®-M33 core as the trusted domain (TD) CPU (see Figure 2. Platform logical scope for details).

For more information on RIF, refer to [RM], section *Resource isolation framework (RIF)*.

3.2.4 *Hardware protections* package of SFRs

This section describes how the *Platform* fulfils the requirements of the *hardware protections* package of *SFR*s.

3.2.4.1 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 countermeasures against physical attacks:

- ROM code execution hardening through redundancy checks and applying timing jitter.
- Detection of transient perturbation attacks targeting cryptographic functions within SAES and PKA private operations.
- Detection of unauthorized modifications to sensitive data stored in fuses.
- Prevention of information leakage via electromagnetic emissions and power consumption during AES operations (in SAES) and private key cryptography (in PKA).

Although the tamper-detection and response hardware of the *Platform* is not activated by default, the integrator can enable it. Enabling these features does not affect the secure initialization of the *Platform SFR*, as described in Section 3.2.1.2: Secure initialization of Platform. For detailed information on the optional anti-tamper features available on the *Platform*, refer to [SG], section 4.2.1.

4 Mapping and sufficiency rationales

4.1 SESIP3 sufficiency

Table 12. SESIP3 sufficiency

Assurance class	Assurance families	Covered by	Rationale
ASE: Security Target evaluation	ASE_INT.1: ST Introduction	Section 1	The ST reference is in the title, the TOE reference in the Section 1.2: Platform references, and the TOE overview and description in Section 1.4: Platform component functional overview and description.
	ASE_OBJ.1: Security requirements for the operational environment	Section 2	The objectives for the operational environment in Section 2: Security objectives for the environment refer to the guidance documents.
	ASE_REQ.3: Listed security requirements	Section 3.2	All *SFR*s in this ST are taken from [SESIP]. *Verification of Platform identity* is included. ~~*Secure update of the Platform*~~ is not included (justification in ALC_FLR.2).
	ASE_TSS.1: TOE Summary specification	Section 3	All *SFR*s are listed per definition, and for each *SFR*, the implementation and verification are defined in Section 3.2 .
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 *SFR*s	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 and material provided to the evaluator	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	N/A(1)	The *Platform* evaluator performs penetration testing to confirm that the potential vulnerabilities cannot be exploited in the TOE operational environment. Penetration testing is performed by the *Platform* evaluator assuming the attack potential as *Enhanced-Basic*.

1. *A vulnerability analysis is performed by the Platform evaluator to ascertain the presence of potential vulnerabilities.*

4.2 Conformance mapping of SESIP profile for PSA Certified Level 3

This section refers to the claims and activities within the *SESIP* evaluation scope to demonstrate sufficiency by *SESIP* methodology.

The following table shows the conformance mapping relative to [SP2], with focus to *PSA Certified Level 3 RoT Component*.

Table 13. SESIP Profile for PSA Certified Level 3 sufficiency

Package claimed	Security Functional Requirements	Reference	Inclusion for RoT Component
Base SP *SFR*s	Verification of *Platform* identity	Section 3.2.1.1	Mandatory
	Verification of *Platform* instance identity	Not claimed	Optional
	Attestation of *Platform* genuineness	Not claimed	Optional
	Secure initialization of *Platform*	Section 3.2.1.2	Optional
	Attestation of *Platform* state	Not claimed	Optional
	Secure update of *Platform*	Section 3.2.1.3	Mandatory
	Physical attacker resistance	Section 3.2.4.1	Mandatory
	Software attacker resistance: Isolation of *Platform*	Not claimed	Optional
	Software attacker resistance: Isolation of *Platform* (between PSA-RoT and application RoT services)	Section 3.2.3.1	Optional
	Cryptographic operation	Section 3.2.2.1	Optional
	Cryptographic random number generation	Section 3.2.2.3	Optional
	Cryptographic key generation	Not claimed	Optional
	Cryptographic key store	Section 3.2.2.2	Optional
Optional *SFR*s	Secure debugging	Section 3.2.1.5	Optional

5 Reference documents

The following table specifies the documents referenced from within this document. They include standards used as reference for the *Platform* evaluation (evaluation documents), and the *Developer* documents.

Table 14. Reference documents

Reference	Definition
Evaluation documents / Standards	
[SESIP]	*Security Evaluation Standard for IoT Platforms (SESIP)*, version 1.2 (July 2023), GlobalPlatform, GP_FST_070
[SP1]	*SESIP Profile for Secure MCUs and MPUs*, version 1.1 (May 2025), GlobalPlatform, GPT_SPE_150
[SP2]	*SESIP Profile for PSA Certified RoT Component Level 3*, version 1.0, JSADEN018
[NIST]	NIST Special Publication (SP) 800-90B (Draft), *Recommendation for the Entropy Sources Used for Random Bit Generation*, January 2018
***Developer* documents**	
[SG]	SG0052, *STM32MP2xxC/xxF security guidance for SESIP level 3 certification*, STMicroelectronics
[RM]	RM0506, *STM32MP21xx advanced Arm®-based 32/64-bit MPUs*, STMicroelectronics RM0457, *STM32MP25x/23x advanced Arm®-based 32/64-bit MPUs*, STMicroelectronics
[DS]	DS14556, *Arm® based Cortex®-A35 1.5 GHz + Cortex®-M33 MPU with TFT, 2× USB 2.0, crypto*, (datasheet of STM32MP21xC/xF), STMicroelectronics DS14634, *Arm® based dual Cortex®-A35 1.5 GHz + Cortex®-M33 MPU, AI, 3D GPU, video decoder, TFT/DSI/LVDS, USB 2.0, crypto*, (datasheet of STM32MP23xC/xF), STMicroelectronics DS14284, *Arm® based dual Cortex®-A35 1.5 GHz + Cortex®-M33 MPU with AI, 3D GPU, video encoder/decoder, TFT/DSI/LVDS, USB 3.0, PCIe®, crypto*, (datasheet of STM32MP25xC/xF), STMicroelectronics
[ES]	ES0628, *STM32MP211x/3x/5x device errata*, STMicroelectronics ES0598, *STM32MP25x/STM32MP23x device errata*, STMicroelectronics
[PSIRT]	https://www.st.com/content/st_com/en/security/report-vulnerabilities.html

Revision history

Table 15. **Document revision history**

Date	Version	Changes
03-Apr-2026	1	Initial release.
10-Jun-2026	2	Revision of the referenced security guidance document updated to *“1”*.
01-Jul-2026	3	Revision of the referenced security guidance document updated to *“2”*. In Section 5: Reference documents, the [SESIP] reference document number corrected to GP-FST-070.

Glossary

BHK Boot hardware key - AES encryption or decryption key stored in backup registers, erased upon tamper event, and not readable or writable after boot by the application.

Developer Entity who developed the device submitted for certification. Here STMicroelectronics.

DHUK Derived hardware unique key

HUK Hardware unique key - a generalization of DHUK and RHUK terms

Integrator Entity who develops a product that uses the device submitted for certification.

IoT Internet of things

MCU Microcontroller unit

MPU Microprocessor unit

NSPE Nonsecure processing environment

NVM Nonvolatile memory - a memory that retains its contents even when powered down, such as flash memory, PCM, OTP memory, or EEPROM

PCI Payment card industry

PKA Public key accelerator

Platform A part of physical device and its software to undergo evaluation of compliance with security requirements. Also called target of evaluation (TOE).

POI Point of interaction

POS Point of sale

PTS Pin transaction security

RHUK Root hardware unique key

RoT Root of trust

SAR Security assurance requirement (*SARs* when used in plural)

SESIP Security evaluation standard for IoT platforms

SFR Security functional requirement (*SFRs* when used in plural)

SP SESIP profile

SPE Secure processing environment

TD Trusted domain

TOE Target of evaluation

Contents

List of tables

List of figures

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.