Secure Target Lite for HX6538-C (WE2)

WiseEye2™ AI Processor

Rev. 1.0 *Nov 2024*

Himax Technology, Inc.
http://www.himax.com.tw

Revision History

Nov 2024

Version	Date	Description of changes
V1.0	2024/11/20	Create ST Lite.

List of Contents

List of Figures

List of Tables

1. Introduction

The Security Target describes the Platform (in this chapter) and the exact security properties of the Platform that are evaluated against [SESIP GP] (in chapter “Security Functional Requirements”) and that a potential consumer can rely upon the product upholding if they fulfill the objectives for the environment (in chapter “Security objectives for the operational environment”).

1.1 SESIP Profile Reference

Reference	Value
PP Name	SESIP Profile for PSA Certified Level 2
PP Version	V1.0 REL 03
Assurance Claim	SESIP Assurance Level 2 (SESIP 2)
Optional and additional SFRs	Secure Debug, Secure Encrypted storage (internal storage), Secure External Storage

Table 1 : SESIP Profile Reference

1.2 Platform Reference

Reference	Value	
Platform Name	HX6538	
Platform Version	C	
Platform Identification	Chip name and version	HX6538-C
	PSA-RoT name and version	Fork of TF-M open-source version 1.7.0. Release Himax source version 1.8
Platform Type	Arm Cortex-M55 microcontroller, running Trusted Firmware-M and Himax proprietary bootrom/bootloader as platform root of trust.	
Trusted Subsystem Identification	N/A	
Trusted Sub-system Certification	N/A	

Table 2 : Platform Reference

1.3 Included Guidance Documents

Reference	Name	Version
[TFM]	Trusted Firmware-M Documentation https://tf-m-user-guide.trustedfirmware.org/index.html	V1.7.0
[PSASTORAGE]	PSA Secure Storage API https://arm-software.github.io/psa-api/storage/	Storage API 1.0.0
[PSACRYPTO]	PSA Crypto API https://arm-software.github.io/psa-api/crypto/	Crypto API 1.0.1
[PSAATTESTATION]	PSA Attestation API https://arm-software.github.io/psa-api/attestation/	Attestation API 1.0.2
[HX-DS]	HX6538-C_DS_preliminary_v04, 07-2024	V04
[HX-AN]	HX6538-C_AN_v02_0410_preliminary, 10-04-2024	V02
[UG-PG]	WE2 Programming Guide, v.1.5, 17-05-2024	V1.5
[UG-KP]	WE2 Key Provisioning, v1.5, 10-04-2024	V1.5

[UG-OTA]	WE2 OTA v2.2.0 User Guide, v2.2.0	V2.2.0
[UG-SD]	WE2 Secure Debug User Guide, v1.4, 01-02-2024	V1.4
[UG-API]	WE2_TF-M_S_API_v0.5	V0.5
[UG-SG]	WE2 Security Guidance, Rev 1.8, July 2024	V1.8
[UG-SEC-CLI]	WE2 Security CLI User Guide, v1.0, 13-12-2023	V1.0
[SESIP-PP-L2]	JSADEN012 SESIP Profile for PSA Certified™ Level 2, v1.0 REL 03	v1.0 REL 03
[SESIP]	GlobalPlatform Technology, Security Evaluation Standard for IoT Platforms (SESIP) Methodology, GP_FST_070, v1.2, 07-2023	v1.2, 07-2023
[SESIP ST]	Secure Target for HX6538-C (WE2) WiseEye2 AI Processor, Preliminary Rev. 2.8, Nov 2024	Rev 2.8

Table 3 : Guidance Documents

1.4 Platform Functional Overview and Description

1.4.1 Platform Type

HX6538-C is an extreme low power, high performance microcontroller designed for battery powered Endpoint AI applications. Dual ARM Cortex M55 CPU cores with Helium vector and floating-point extensions and an ARM Ethos U55 micro NPU core to accelerate convolution operation of neural network model. Power Management unit to fulfill multi-layer power functions. Security features fulfill with ARM TrustZone, CryptoCell-312 crypto hardware and Physical Unclonable Function(PUF).

1.4.2 Physical Scope

The hardware is a System-on-Chip.
The major scope for ToE are dual ARM Cortex M55 CPU cores. (big 400Mhz, little 100Mhz) with security extension, SAU/IDAU, Memory Protection Unit(MPC), Peripheral Protection Controller(PPC), Memory Protection Unit(MPU) and Master Security Controller(MSC) to combine and cooperate with TrusTzone. Crypto hardware CrytpoCell-312 is used to accelerate secure operation such as secure boot, secure update and cryptographic operations. PUF is a key storage OTP and it also support TRNG. The TOE source code would be released on the Himax FTP link after customer is signed an NDA with Himax.

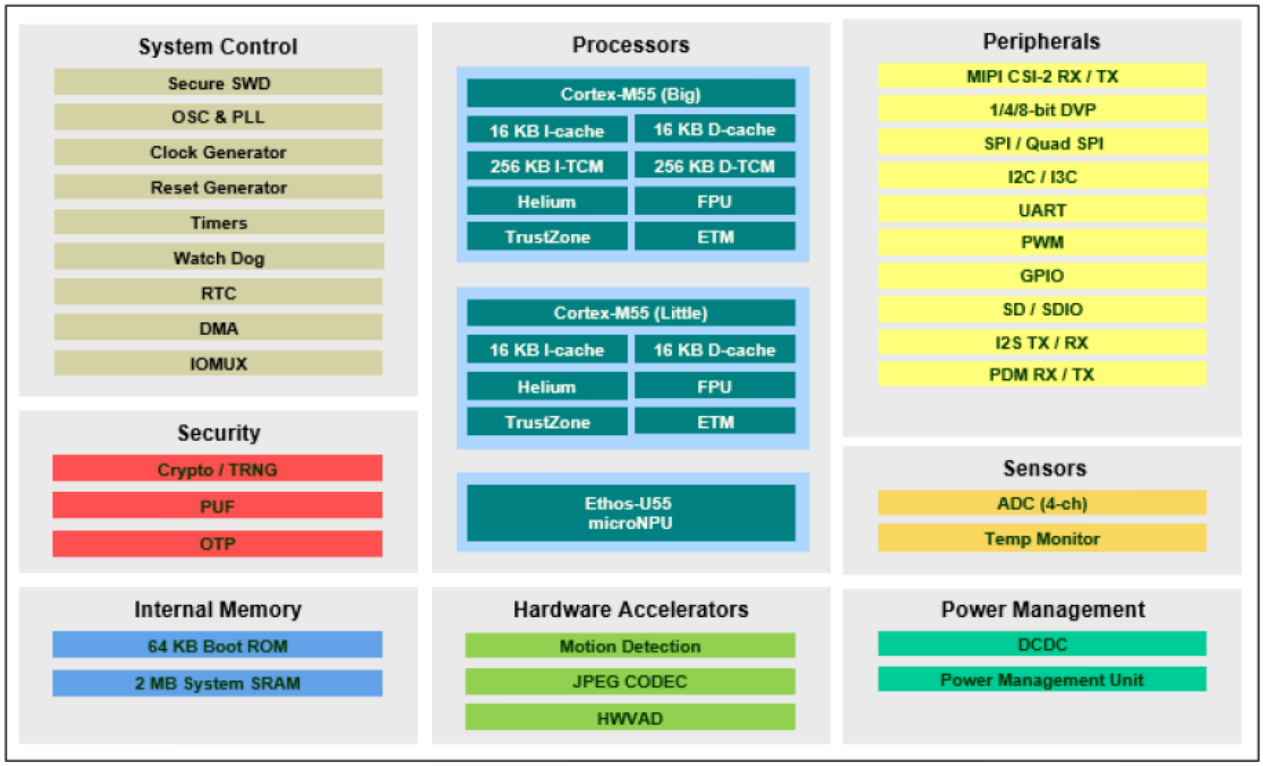

Figure 1 : HX6538-C block diagram

Type	Name	Release version	Form of delivery	SHA256
IC hardware	HX6538-Cxx	Rev C	Silicon Chip and Chip ROM	None
BootRom	WE2_BootRom	SVN:175281	On chip ROM firmware	8ab716828d74647eee37ccdd9bc3f116f6e248a5d3db93419b312a49e1f17492
BootLoader	WE2_BootLoader	v 2.10	Software package with zip file	b8eba40600c428d81c927c4ff235fc4ff2a72e8fa961dccd3c31ea04fd587e6b
2ndBootLoader	WE2_2ndBootLoader	v 2.3	Software package with zip file	a8d02af84343c2d0b68e832f56f59b6b3454a38c8d4d755d1641f4e0e1af4298
Application FW	WE2_PSA_TF-M_S	v 1.8	Software package with zip file	268b61095d19aca00a9241e7d2e273b1bf42e9c3eb6a1fa5d4224d932f3514d7
	WE2_PSA_TF-M_NS	v 1.4	Software package with zip file	eb8d7cbf5c1b8906731a80fdaacd1a71e08dd5a2480aee1c5c39af4b3eab2472

Table 4 platform source deliverables

1.4.3 Logical scope

The scope for Target of Evaluation (TOE) comprised of TF-M compliant firmware with App-Rot, PSA-RoT service , secure partition manager and secure drivers. Below red box

represents the TOE parts in Figure2.

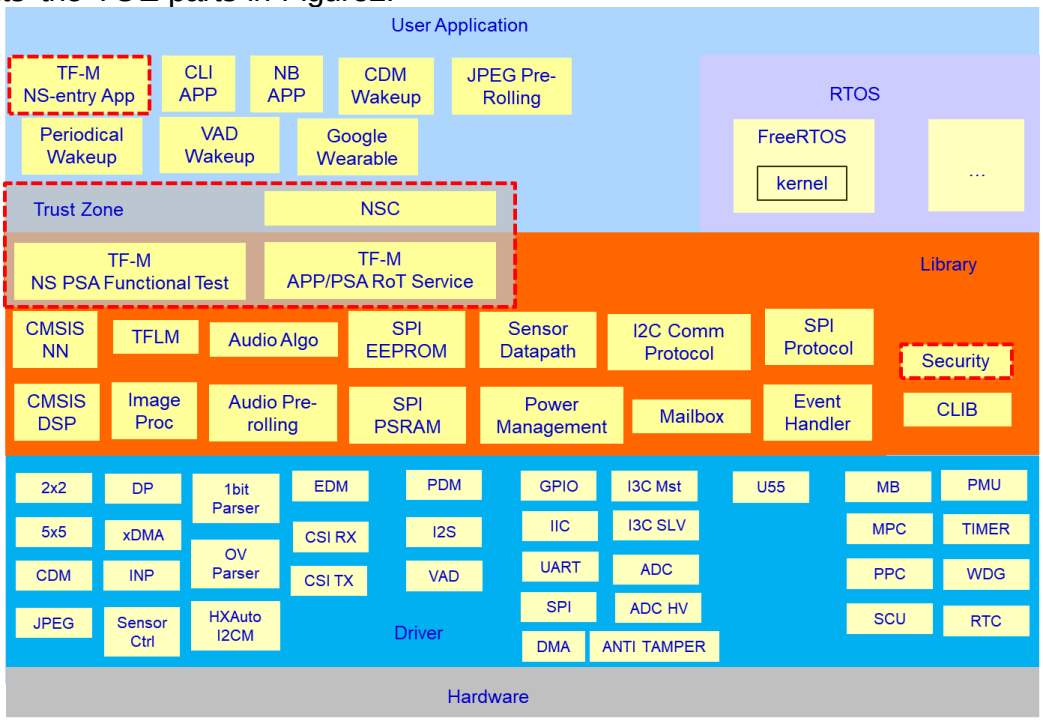

Figure 2 : Software architecture for TOE

The Chip security evaluation scope includes the following Secure Processing Environment PSA-RoT elements as described below :

Immutable Platform Root of Trust :

BootRom, OTP, keys store in OTP for secure boot verification, cryptocell-312 security hardware, hardware based security lifecycle. The immutable part is in secure mode and NSPE can't access above security functions.

Updateable Platform Root of Trust :

Bootloader, TF-M code(secure partition manager, App-Rot, PSA-RoT), OTA Loader. Above software are running in secure mode, NSPE can't access SPE resource directly. If NSPE need to use SPE resource, NSPE need to invoke non-secure callable(NSC) interface to access secure resource.

1.4.4 Usage and Major Security Features

HX6538-C secure features and usage are describe as below :

NSPE/SPE :

Secure process environment(SPE) and non-secure process environment(NSPE) are divided by TrustZone security setting. SAU/IDAU and MPC to config memory as secure or non-secure attribute. And PPC config peripheral device as secure or non-secure. MSC config master device such as U55, DMA with secure and non-secure attribute.

In the platform, the platform set secure attribute by MPC and the secure attribute configuration comes from secure memory layout which is store in himax memory descriptor. The himax memory descriptor is also verified in BootRom to check it's authenticity and integrity. The platform set SAU/IDAU configuration, PPC and MSC in TF-M secure application. The platform set above setting to fulfill secure installation.

Firmware Image :
Secure/Non-Secure firmware is signed by CM/DM key owner in operational environment. In secure boot stage, the firmware image would be verified by CM/DM public key. After the firmware image has been verified, it means that the firmware's authentication belongs the key owner.

Secure boot :
The secure boot provides a secure foundation for customer firmware. HX6538-C bootloader is started from SPE to provides authentication, decryption, integrity verification and version checking for customer firmware on installation and boot/reset. And set secure resource permission in TF-M.

Secure update :
The secure OTA is executed in HX6538-C firmware and loader. The secure OTA uploader provides the authentication, decryption, integrity verification and firmware version checking for customer firmware before upgrade firmware to Flash memory.
Below are the tasks for secure OTA :
- OTA APP in NSPE invokes PSA firmware update (FWU) service in SPE to initial and setup OTA process in SPE.
- OTA process will initial drivers and set whole memory with secure attribute.
- OTA process does firmware version checking, firmware decryption and signature verification.
- OTA process can do the re-encryption with hardware unique key (HUK) (optional).
- OTA process updates firmware to Flash memory.

Key storage :
Crypto keys are store in PUF OTP cell. Keys are listed in section 3.2.12 Cryptographic KeyStore

Life Cycle States(LCS) :
HX6838-C support 4 life cycle state. Chip initial state is Chip Manufacture (CM).
Chip Manufacture (CM) :
- In this stage, it doesn't support secure boot and secure update.
- Chip debugging is enabled.
- No keys are provisioned into OTP.
- After CM keys are provisioned, LCS change to DM.

Device Manufacture (DM) :
- In this stage, it supports secure boot and secure update because CM keys were provisioned.
- Chip debugging is enabled through secure debug to enable debug function. DM request secure debug from CM. Thus, it can debug secure and non-secure firmware.
- After DM keys are provisioned, LCS change to SM.

Secure Enable (SE) :
- In this stage, it supports secure boot and secure update by CM owner or DM owner because CM and DM keys were provisioned.

- Chip debugging is disabled. If developer want to enable debug ability, developer need to request from CM or DM. If request from CM, it can debug secure and non-secure firmware. If request from DM, it can debug only non-secure firmware.
-

Return Merchandise Authorization (RMA) :
- A terminal state for devices that are returned to the ICV for analysis of fatal failures.

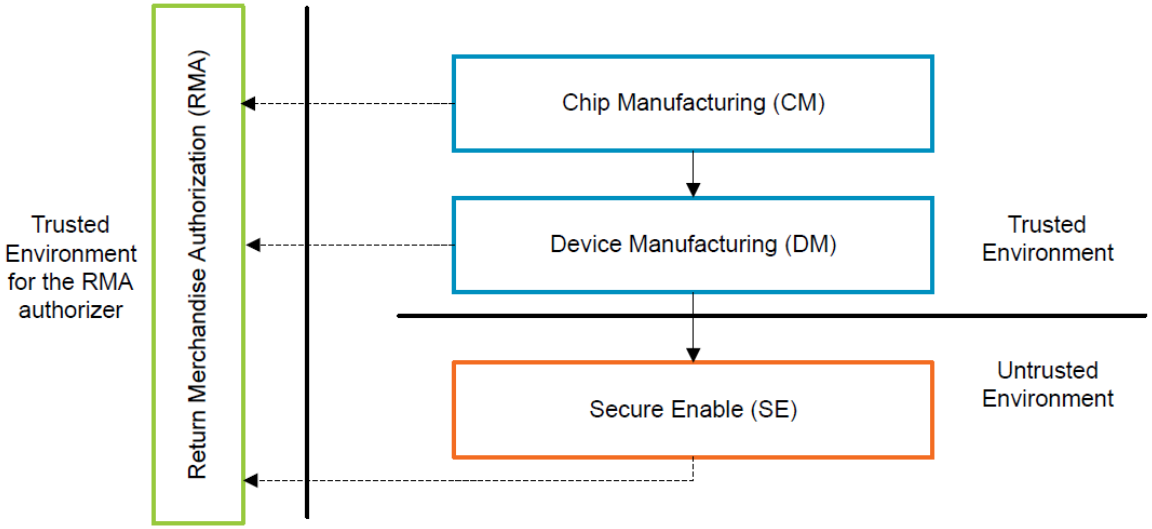

Figure 3 : Security life-cycle states

Crypto processor :
Arm CryptoCell Hardware Accelerator CC312.
Cryptographic functions services for SPE and NSPE applications.

Attestation :
The attestation on the platform is based on signature verification. System functions (such as platform identify, platform instance identity and LCS status...etc.) can be used after the firmware image is verified by hash of public key.

1.4.5 Required Hardware/Software/Firmware

The TOE execute TF-M framework also need other minimum drivers to support it. Below list the required hardware, software.
Hardware : SPI, UART, Flash, I2C.
Software :
- SPI I/F connect to CMSIS flash read/write API for burning flash data.
- UART I/F connect to CMSIS uart read/write API for console log.
- I2C I/F is used in OTA loader to communicate with host.

2. Security Objectives for the operational environment

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

ID	Description	Reference
KEY_MANAGEMENT	CM : Cryptographic keys are stored and generated in himax HSM. DM : Cryptographic keys are stored and generated in ODM's HSM and managed by themself. Keys are packed by CMPU/DMPU tool and then burn into platform	[UG-KP] in section 1.1, 2.1 [UG-SG] in section 3.2.4
TRUSTED_USERS	CM : Himax release licensed card for trusted users. Each card is created by public key cryptography. For himax internal only. Developers insert this card to a card reader and it would be link to Himax internal server to check and verify the card identity. DM : ODM can follow Himax method to use licensed card or their own way.	[UG-KP] in section 1.1, 2.1 [UG-SG] in section 3.2.4
UNIQUE_ID	The unique ID is SOC ID which based on HUK and hash of public key (HBK). Both of HUK and HBK are provisioned during key provisioning stage.	[HX-ST] in section 3.2.2, [UG-KP] in section 1.2~1.4, 2.2~2.4 [UG-SG] in section 3.2.4

Table 5 Security Objectives for the operational environment

3. Security Requirements and Implementation

3.1 Security Assurance Requirements

The claimed assurance requirements package is SESIP2 as described in Section 4.1.

3.1.1 Flaw Reporting Procedure

In accordance with the requirement for a flaw reporting procedure (ALC_FLR.2), including a process to report a flaw and generate any needed update and distribute it, the developer has defined the following procedure:
Reporting: customer report their issues to Himax technical support web.
Evaluation:

- Himax FAE would evaluate and reproduce the error condition.
- For security part: Himax FAE would run security CLI and debug it from error code & log(refer [UG-SEC-CLI]).

Solution: Himax would supply a solution to customer by change FW or IC(depending on analysis result.)
Communication: Himax FAE would communicate to customer to check customer's feedback by customer's e-mail or phone number which are filled in technical support website.
Technical support URL : https://www.himax.com.tw/support/technical-support/

3.2 Base PP Security Functional Requirements

As a base, the platform fulfils the following security functional requirements:

3.2.1 Verification of Platform Identity

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

Conformance rationale:

HX6538-C provides a read chip version API - veneer_sys_get_version to get 64 bits chip version to represent its identification.
The chip version is read only register.

Self-assessment:
TOE log:

```
Select test function
version1 = 0x8538000c
version2 = 0x8538000c
```

3.2.2 Verification of Platform Instance Identity

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

Conformance rationale:

HX6538-C provides a read SoC id API - psa_hx_security_get_socid to get a SOCID string which is 256 bits value to represent every chip unique identification.
The SOCID is derived from HUK and Hash of public key. HUK and Hash of public key would be checked keys zero bits. If the keys counted zero bits don't match to zero bit value which store in OTP, the derived flow would fail. Every HX6538-C is unique because of their unique HUK. Therefore, HX6538-C has different SOCID in CM and DM LCS.

Self-assessment:
For different key, the SOC ID is different.
TOE log:

```
SocID = 712b5d1b1854e80df885f5df40461bb360988237b3e5c226d96323121a47c18e
```

3.2.3 Attestation of Platform Genuineness

The platform provides an attestation of the "Verification of Platform Identity" and "Verification of Platform Instance Identity", in a way that ensures that the platform cannot be cloned or changed without detection.

Conformance rationale:

The attestation flow is start from NS user invoke psa_initial_attest_get_token. The sign and verify key pair is ECDSA key. The ECDSA private key is provisioned and generated in CMPU provision key stage. The public key could be loaded from NS side for user to verify the attestation token. NS users send a nonce and this API would return a token with signature and hashed identity value. NS users perform verification with nonce and signature to perform signature verification.

Self-assessment:
Refer [UG-PG] section Execute WE2 – Console Log PSA Functional Tests

- INIT_ATTEST_TestLog

3.2.4 Secure Initialization of Platform

The platform ensures its authenticity and integrity during platform initialization. If the platform authenticity or integrity cannot be ensured, the platform will go to a state where no other operation except optionally Secure Update of Platform can be performed.

Conformance rationale:

HX6538-C secure initialization starts from secure boot. Below description of secure boot follow three major stage. The first flow is secure boot stage, it describes whole boot flow. Second is signature verify stage, it describes what verification do. Third is decrypt image stage, it describes images integrity.

Secure Boot flow:
The secure boot flow is.

- Chain of trust from the Boot ROM.(bootrom -> bootloader -> application)
- Verify integrity and authenticity
  - Boot Rom verify and decrypt Bootloader image and then jump to Bootloader.
  - Bootloader verify and decrypt APP image and then jump to APP.
- If bootrom or bootloader verification is failed, secure boot would jump to load OTA loader. OTA loader would do secure update. (refer to 3.2.6 for detail)
- If boot rom secure boot jump to OTA loader fails, it starts to run boot from others by I2C/SPI/UART.
  - User could use those interfaces to send their boot image. If the image verification is done, boot code would start to execute the received image.
  - This is the final backup mechanism for boot. If bootloader and OTA loader are failure.
  - For some customers, they can't burn flash after the chip leave production line.

Signature verify description:
In this stage, the verification behavior includes signature verify app image and signature verify 2nd bootloader.

- Life cycle status (LCS):
  - Protects security assets in different LCS.
  - DM stage only verify secure application which is signed by CM private key.
  - SE stage verify secure application which is signed by CM private key and non-secure application which is signed by DM private key.
- Authentication:
  - verify image by RSA2048/3072
- Anti-rollback:
  - check Image version with NV counter in OTP.

Decrypt image description:

- Confidentiality: AES-CTR-128 to decrypt image.
- Integrity:
  - calculate hash of image by sha256
  - secure boot compare hash of image to hash which is store in image header.

Self-assessment:
Every time TOE perform boot is fulfill secure initialization of platform. If boot success or failed, this means this SFR matched.

3.2.5 Attestation of Platform State

The platform provides an attestation of the state of the platform, such that it can be determined that the platform is in a known state.

Conformance rationale:

The attestation flow is start from NS user invoke psa_initial_attest_get_token. The sign and verify key pair is ECDSA key. The ECDSA private key is provisioned and generated in CMPU provision key stage. The public key could be loaded from NS side for user to verify the attestation token. NS users send a nonce and this API would return a token with signature and hashed platform state value. NS user perform verification with nonce and signature to perform signature verification.

Self-assessment:
Refer [UG-PG] section Execute WE2 – Console Log PSA Functional Tests
- INIT_ATTEST_TestLog

3.2.6 Secure Update of Platform

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

Conformance rationale:

HX6538-C secure update starts from NSPE or SPE. When the FWU service receive PSA call : psa_fwu_start, HX6538-C would jump into OTA loader to do firmware upgrade. Below stages describes the secure update flow.

Host trigger start OTA command stage:
- Support invoke OTA loader from secure/non-secure environment.
- If OTA is triggered from NSPE, it would through an OTA NSC call.

Receive image stage:
- Update image comes from I2C interface.
- Receive update image into secure SRAM. (all memory would be set as secure attribute in OTA loader initial stage.)

Signature verify stage:
- Authentication:
  - verify image by RSA2048/3072
- Anti-rollback:
  - check Image version with NV counter in OTP.

Decrypt image stage:
- Confidentiality: AES-CTR-128 to decrypt image.
- Integrity :
  - calculate hash of image by sha256
  - secure boot compare hash of image to hash which is store in image header.

Re-encryption stage: Optional (For higher security level)
- Use derived HUK to re-encrypt image.
- After re-encryption is done, every image is different in each chip.

Write update image to flash stage:
- After check the authenticity, confidentiality and integrity, OTA loader burn image into flash and reboot.

Self-assessment:
Refer [UG-OTA] to perform OTA update. Once the upgrade is succeeded, the executable firmware is changed.

3.2.7 Software Attacker Resistance: Isolation of Platform (between SPE and NSPE)

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

Conformance rationale:

The platform provides isolation between NSPE and SPE through the TrustZone support of the HW and using the TF-M firmware to enable isolated secure services running in the SPE.

Self-assessment:
Refer [UG-PG] section Execute WE2 – Console Log PSA Functional Tests for all tests. All the tests come from NS side through NSC. NS data are store in NS side.

3.2.8 Software Attacker Resistance: Isolation of Platform (between PSA-RoT and Application Root of Trust Services)

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

Conformance rationale:

The platform runs Trusted Firmware-M(TF-M) with isolation level 2 which separate SPE into PSA-RoT and App-RoT by TrustZone privileged/privileged mode.
Privilege mode(PSA-RoT) : CPU can access all system resources such as register, timer ... etc. in this PRoT.
Unprivileged mode (App-RoT) : CPU **can't** access all system resources such as MSR, MRS instruction, register and some privilege hardware in ARoT.

Self-assessment:
Refer [TFM] It's based on TF-M framework to do this.

3.2.9 Cryptographic Operation

The platform provides the application with Operations in Table 6 functionality with algorithms in Table 6 as specified in specifications in Table 6 for key lengths described in

Table 6 and modes described in Table 6.

Conformance rationale:

Algorithm	Operations	Specifications	Mode	Key length
AES	Encryption/Decryption	NIST FIPS 197 (AES) NIST SP800-38A (ECB, CBC, CFB, OFB, CTR)	ECB, CBC, CTR, OFB,	128, 192, and 256 bits.
	Message authentication and key derivation	NIST SP800-38B NIST SP800-108r1	CMAC, CBC-MAC	128, 192, and 256 bits.
	Authenticated encryption/decryption with additional data	NIST SP800-38C (CCM)	CCM	128, 192, and 256 bits
		NIST SP800-38D (GCM)	GCM	128, 192, and 256 bits
Hash	Secure hashing Keyed hashing for HMAC	NIST FIPS 180-3	SHA2 (SHA224, SHA256)	n/a
HMAC	Message authentication	RFC2104	SHA2 (SHA224, SHA256)	n/a
RSA PKCS#1	Encryption Decryption Signature generation Signature verification	PKCS #1 v2.1 & v1.5 Public-Key Cryptography Standards RSA Cryptography Specifications	PSS, OAEP	2048, 3072, and 4096 bits.
Elliptic curve digital signature	Signature generation Signature verification	NIST FIPS 186-4 SEC 2	n/a	NIST curves secp256r1, secp521r1, secp256k1, secp384r1
CMAC	Key derivation	NIST SP800-108 Recommendation for Key Derivation Using Pseudorandom Functions	CMAC	128, 192, 256 bits
TRNG	Random number generator	Compliant with NIST SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation.	n/a	n/a

Table 6 : CryptoCell-312 supported algorithms

Self-assessment:
Refer [UG-PG] section Execute WE2 – Console Log PSA Functional Tests: CRYPTO_TestLog.

3.2.10 Cryptographic Random Number Generation

The platform provides the application with a way based on PUF entropy pool to generate random numbers to as specified in NIST SP 800-90B.

Conformance rationale:

The platform provides TRNG API psa_generate_random to get TRNG number from PUF. This psa_generate_random has been integrated to Himax PUF function.

Self-assessment:
Refer [UG-PG] section Execute WE2 – Console Log PSA Functional Tests: CRYPTO_TestLog

3.2.11 Cryptographic Key Generation

The platform provides the application with a way to generate cryptographic keys for use in cryptographic operations in Table 7 as specified in specifications in Table 7 for key lengths described in Table 7.

Conformance rationale:

ID	Algorithm	Specifications	Key length
RSA	RSA	PKCS#1 v2.1 & v1.5	2048 and 3072 bits.
ECC	ECDSA	SEC 2 FIPS 186-4	NIST curves secp256r1, secp521r1, secp256k1, secp384r1.

Table 7 : CryptoCell-312 supported key generation

Self-assessment:
Refer [UG-PG] section Execute WE2 – Console Log PSA Functional Tests: CRYPTO_TestLog.

3.2.12 Cryptographic KeyStore

The platform provides the application with a way to store cryptographic keys such that not even the application can compromise the authenticity, integrity, confidentiality of this data. This data can be used for the cryptographic operations listed in section 3.2.9 Cryptographic Operation.

Conformance rationale:

Keys are provisioned into PUF OTP that is provided by CMPU and DMPU burn key flow. The platform executes the CMPU or DMPU application to provision the CM or DM key package which are generated from host side CMPU or DMPU key generation tool. Himax would provision CM keys before chip release. ODM developers or evaluators provision DM key by themselves.

Self-assessment:
Refer [UG-KP] section 2.2 The OEM Asset-Packaging PC Tool (DMPU).

3.3 Optional Security Functional Requirements

3.3.1 Secure Debug

The platform only provides certificate which is stored in flash and is used for authentication as specified in secure debug document [UG-SD, WE2 Secure Debug User Guide, PKCS #1 v2.1] with debug functionality.

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

Conformance rationale:

The platform enables secure debug by verifying secure debug certificate. The secure debug certificate is stored in flash and the certificate would be verified in booting stage.
Secure debug can enable limited serial wire debug (SWD) ability by CM or DM key owner.
For CM key owner, the secure debug can enable secure and non-secure debug ability.
For DM key owner, the secure debug can enable only non-secure debug ability.
The debug functions contain SoC-600 APB Access Port, SoC-600 AHB Access Port, SDC-600, ESS600 CTI and M55-Big/Little enable. Each of above has secure and non-secure debug ability.
The detail of secure debug function can be referred from [UG-SD P.6. Table.1].

Self-assessment:
Refer [UG-SD], [UG-PG] SWD Debug. Once it success, MCU Link can be used to debug TOE. Others means it failed.

3.3.2 Secure Encrypted Storage (internal storage)

The platform ensures that all data stored by the application, except for data store in ROM, OTP and non-secure domain, is encrypted by AES-CCM (followed by standard NIST SP800-38D) as specified in [HX-DS] with a platform instance unique key of key length 128 bits.

Conformance rationale:

Secure Encrypted Storage is provided by the TF-M PSA Protected Storage (PS) service, which provides functionality to store generic data in non-volatile memory inside the SPE.
The service is usually backed by hardware isolation of the SRAM2 access domain and, in the current version, relies on hardware to isolate the SRAM2 area from non-secure access.
In absence of hardware isolation, the secrecy and integrity of data is still maintained.
The PS service implements an AES-CCM based AEAD encryption policy, as a reference, to protect data integrity and authenticity.
The PSA PS uses a 128 bits key which is derived from a Hardware Unique Key (HUK) stored in the PUF OTP cell. (HUK is 256bits, in PS service use 128bits only.)
The HUK is protected from PUF and only CryptoCell-312 can read it when executing AES algorithm.
The PS service would invoke encryption/decryption in object system module and then invoke crypto algorithm in crypto partition. After the encryption/decryption are done, it invokes ITS API to write/read encrypted data into SRAM2.

Self-assessment:
Refer [UG-PG] section Execute WE2 – Console Log PSA Functional Tests: PS_TestLog.
Encryption and authentication algorithm execute in PS service. The PS service would perform encryption and authentication to data and then store data through ITS service API.

3.3.3 Secure External Storage

The platform ensures that all data stored outside the direct control of the platform, except for data wrapped in raw mode, is protected such that the authenticity, integrity, confidentiality and binding to the platform instance, versioning (with version number in image gen configuration is ensured).

Conformance rationale:

The boot image is store in flash. The platform flash is external to the platform. The image can be store in BLP, BLW, BLW_REENC or RAW mode. The 4 modes are created by image generation tool. Each mode (BLP, BLW, BLW_REENC) of image is composed with a secure cert and concatenation with binary data. RAW mode partition of image is composed with binary data only and without certificate.
BLP mode means binary data is signed without data encryption.
BLW mode means binary data is encrypted and signed.
BLW_REENC mode means binary data in BLW mode and it will be re-encrypted in TOE runtime stage. This would use unique key to do firmware re-encryption.
RAW mode means plain data is stored in flash without data encryption and signature. Application neural network model data are wrapped into RAW mode.
Firmware binary store as BLP, BLW, BLW_REENC mode into image. In boot stage (boot rom and bootloader), the image data would be loaded from flash, and it would be verified (authenticity), compared with hash content (integrity) and decrypted (confidentiality – BLW mode) by secure boot API.
The encryption and decryption algorithm use AES-CTR-128. The hash algorithm use SHA256. The signature verification algorithm use RSA2048.
For data wrapped into raw mode, the binary data is only allowed to move into NSPE.
If developer want to invoke SPI read/write non-secure callable API (veneer_spi_eeprom_2read, veneer_spi_eeprom_write) to read & write byte array data from/to external flash. The byte array data should use psa_hx_security_boot API to check its authenticity, integrity and confidentiality. But for TOE current design, TOE only need to use SPI read function to interact with psa_hx_security_boot API.

Self-assessment:
Every time TOE perform boot is fulfill this SRF because all images read from flash. If boot success or failed, this means this SFR matched.

3.3.4 Secure External Storage

The platform ensures that data stored outside the direct control of the platform, except for plain data, is protected such that the authenticity, integrity, confidentiality and binding to the platform instance, versioning (with PS object system version).

Conformance rationale:

Secure External Storage is provided by the TF-M PSA Protected Storage (PS) service, which provides functionality to store generic data in non-volatile memory inside the SPE.
This service support writes encryption data to flash or read decryption data from flash.
It also supports same operation to write/read data to/from SRAM2. (Refer 3.3.2)
The PS service implements an AES-CCM based AEAD encryption policy, as a reference, to protect data integrity and authenticity.

The PSA PS uses a 128 bits key which is derived from a Hardware Unique Key (HUK) stored in the PUF OTP cell. (HUK is 256bits, in PS service use 128bits only.)
The HUK is protected from PUF and only CryptoCell-312 can read it when executing AES algorithm.
The PS service would invoke encryption/decryption in object system module and then invoke crypto algorithm in crypto partition. After the encryption/decryption are done, it write/read encrypted data into flash device.

Self-assessment:
Refer [UG-PG] section Execute WE2 – Console Log PSA Functional Tests: PS_TestLog.
Encryption and authentication algorithm execute in PS service. The PS service would perform data read/write to flash.

4. Mapping and Sufficiency Rationales

4.1 Assurance

The assurance activities defined in [PSA-EM-L2] fulfil the SESIP2 activities. In particular, the required source code review, vulnerability analysis and testing to an equivalent of 25 person-days of the [PSA-EM-L2] is applicable.

Assurance Class	Assurance Family	Covered by
ASE: Security Target evaluation	ASE_INT.1 ST Introduction	Section 1 Introduction
	Rationale: ST reference: see Section 1.1 Platform reference: see Section 1.2 Platform function overview and description: see Section 1.4	
	ASE_OBJ.1 Security requirements for the operational environment	Section 2 Security Objectives for the operational environment
	Rationale: Objectives for the operational environment: see Section 2	
	ASE_REQ.3 Listed Security requirements	Section 3 Security Requirements and Implementation
	Rationale: All security requirement are listed in section 3.	
	ASE_TSS.1 TOE Summary Specification	Section 3 Security Requirements and Implementation
	Rationale: All SFRs are listed per definition, and for each SFR the implementation and rationale are provided in the SFR.	
ADV: Development	ADV_FSP.4 Complete functional specification	Section 1.3 Included Guidance Documents
	Rationale: The guidance documents explained TOE's hardware component, software component and software API for developer to develop TOE. Document in [HX_DS], [TFM]	
AGD: Guidance documents	AGD_OPE.1 Operational user guidance	Section 1.3 Included Guidance Documents

	Rationale: The guidance documents explained how to operate the TOE. Document in [UG-KP], [UG-PG], [HX-AN]	
	AGD_PRE.1 Preparative procedures	Section 1.3 Included Guidance Documents
	Rationale: The guidance documents explained how to prepare hardware and image of the TOE. Document in [UG-PG], [HX-AN]	
ALC: Life-cycle support	ALC_FLR.2 Flaw reporting procedures	3.1.1 Flaw Reporting Procedure
	Rationale: The flaw reporting and remediation process is described in this document.	
ATE: Tests	ATE_IND.1 Independent testing: conformance	Testing carried out by the laboratory
	Rationale: The laboratory shall perform independent test.	
AVA: Vulnerability Assessment	AVA_VAN.2 Vulnerability analysis	Vulnerability and testing carried out by the laboratory
	Rationale: The laboratory shall perform vulnerability analysis and penetration test.	

Table 8 : Assurance Mapping and Sufficiency Rationales

4.2 Functionality

PSA Security Function	Covered by SESIP SFR	Rationale
F.INITIALIZATION	Secure Initialization of Platform	Full coverage by HX6538-C based on bootrom to bootloader.
F.SOFTWARE_ISOLATION	Software Attacker Resistance: Isolation of Platform (between SPE and NSPE)	Full coverage by TrustZone from Cortex-M55 processor and MPC, PPC, MPU and MSC.
	Software Attacker Resistance: Isolation of Platform (between PSA-RoT and Application Root of Trust Services)	Full coverage by using unprivileged/privileged memory separation between ARoTs and the PRoT
	(Optional) Software Attacker Resistance: Isolation of Application Parts (between each of the Application Root of Trust services)	Not provided by TOE. TOE is target on PSA Level2. PSA level 2 doesn't need to isolate each App RoT.
F.SECURE_STORAGE	Secure Encrypted Storage (internal storage)	For secure encrypted storage, it is covered in section 3.3.2
	Secure Storage (internal storage)	Not provided by TOE
	Software Attacker Resistance: Isolation of Platform (between SPE and NSPE)	Full coverage by TF-M isolation to separate NSPE and SPE. NSPE can't access SPE.
	Secure External Storage	Image data store in external flash. Boot rom load boot image from external flash. Covered in section 3.3.3 For Protect Storage Service, it is covered in section 3.3.4
F.FIRMWARE_UPDATE	Secure Update of Platform	Full coverage by HX6538-C based on secure OTA.
F.SECURE_STATE	Software Attacker Resistance: Isolation of Platform (between SPE and NSPE)	Full coverage
	Software Attacker Resistance: Isolation of Platform (between PSA-RoT and Application Root of Trust Services)	Full coverage
	Partially covered by the SFR "Secure initialization of platform" and "Secure update of platform".	Full coverage
	Limited Physical Attacker Resistance	Not provided by TOE. TOE does not have hardware to prevent from physical attack.
F.CRYPTO	Cryptographic Operation	Full coverage by TF-M accessible through PSA Crypto API.
	Cryptographic KeyStore	Full coverage by TF-M accessible through PSA Crypto API.
	Cryptographic Random Number	Full coverage by TF-M accessible through PSA Crypto API.
	Cryptographic Key Generation	Full coverage by TF-M accessible through PSA Crypto API.
F.ATTESTATION	Verification of Platform Identity	Platform identity can be identified by himax get version API.
	Verification of Platform Instance Identity	Platform instance identity can be identified by psa_hx_security_get_socid API.

	Attestation of Platform Genuineness	"Verification of Platform Instance" and "Verification of Platform Instance Identity" are covered.
	Attestation of Platform State	This information is available by life cycle status.
F.AUDIT	Audit Log Generation and Storage	Not provided by TOE. TOE is a resource constrained device which is a limited memory device and doesn't have enough space to store audit log.
F.DEBUG	Secure Debugging	Debug port is locked by default. It can be enabled by secure debug certificate.

Table 9 : Functionality Mapping and Sufficiency Rationales