Arm® CryptoIsland™-300P

Integrated Secure Element Security Target Lite

Non-Confidential

Issue 2.0

Document ID: 107611

Arm® CryptoIsland™-300P
Integrated Secure Element Security Target Lite

Release information

Document history

Issue	Date	Confidentiality	Change
1.0	July 2022	Non-Confidential	Initial version
2.0	July 2022	Non-Confidential	Fixes following certifier comments

Non-Confidential Proprietary Notice

Arm Limited. Company 02557590 registered in England.
110 Fulbourn Road, Cambridge, England CB1 9NJ.
(LES-PRE-20349)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Inclusive language commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used language that can be offensive. Arm strives to lead the industry and create change.

This document includes language that can be offensive. We will replace this language in a future issue of this document. To report offensive language in this document, email **terms@arm.com**.

Contents

1. Introduction

1.1. Terms and abbreviations

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: **https://developer.arm.com/glossary**.

Table 1-1: Terms and abbreviations

Term	Meaning	Type
AP	Application processor	Abbreviation
NVM	Non-Volatile Memory	Abbreviation
MRAM	Embedded Magnetic RAM	Abbreviation
APDU	Application Protocol Data Unit	Abbreviation
SC	Secure component (Secure element)	Abbreviation
SSI	Security Subsystem Interface	Abbreviation
MHU	Message Handling Unit	Abbreviation
LLRAM	Low Leakage RAM	Abbreviation
TRAM	Trusted RAM	Abbreviation
ATU	Address Translation Unit	Abbreviation
MSU	Memory Security Unit	Abbreviation
MPU	Memory Protection Unit	Abbreviation
OTP	One Time Programming	Abbreviation
FSB	First Stage Bootloader	Abbreviation
SSB	Second Stage Bootloader	Abbreviation
FUT	Firmware Update Tool	Abbreviation
NVR	Non-Volatile Registers	Abbreviation
PSI	Platform signal interface	Abbreviation
SoC	System on Chip	Abbreviation
3S	Secure Subsystem in System-on-Chip	Abbreviation

1.2. Related documents

The following table contains a list of referenced standards and other documents.

Table 1-2: Related documents

Ref	Document ID and revision	Publication Date	Document name
1	FIPS 197	November 2001	Advanced Encryption Standard
2	NIST SP 800-38A	December 2001	Recommendation for Block Cipher Modes of operation: Methods and Techniques
3	NIST SP 800-38B	May 2005	Recommendation for Block Cipher Modes of Operation: the CMAC Mode for Authentication
4	FIPS publication 180-4	August 2015	Secure Hash Standard (SHS)
5	NIST SP 800-90A	January 2012	Recommendation for Random Number Generation Using Deterministic Random Bit Generators
6	BSI AIS-31	September 2011	Functionality Classes and Evaluation Methodology for True Random Number Generators
7	NIST SP 800-22	April 2010	A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications
8	SEC 2, Version 2.0	January 27, 2010	Recommended Elliptic Curve Domain Parameters
9	RFC5639	March 2010	Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation
10	SEC 1, Version 2.0	May 21, 2009	Elliptic Curve Cryptography
11	FIPS Publication 186-4	July 2013	Digital Signature Standard (DSS)
12	NIST SP 800-56A, Revision 3	April 2018	Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography
13	NIST SP 800-108	October 2009	Recommendation for key Derivation Using Pseudorandom Functions
14	ISO/IEC 9797-1	2011	Information technology — Security techniques — Message Authentication Codes (MACs)
15	RFC7748	January 2016	Elliptic Curves for Security
16	BSI-CC-PP-0084-2014, version 1.0	January 13, 2014	Security IC Platform Protection Profile with Augmentation Packages
17	Version 1.3	November 2020	External NVM Storage Augmentation Package
18	Version 1.0	February 2016	Security Requirements for Post-Delivery Code Loading
19	BSI-DSZ-CC-PP0117-2022, Version 1.5	February 2022	Secure Sub-System in System-on-Chip (3S in SoC) Protection Profile

2. Security Target Introduction

2.1. Security Target Reference

The Arm CryptoIsland-300P Integrated Secure Element Security Target Lite version is 2.0 is dated July 2022.

2.2. TOE Reference

The Target of Evaluation (TOE) is Arm Integrated Secure Element CryptoIsland-300P, abbreviation: CI-300P, version 1.0.

2.3. TOE Definitions

This Security Target is defined for the Arm® CryptoIsland™-300P Hard Macro, its bootloaders, Firmware Update Tool (FUT) and Runtime Library. The Hard Macro is embedded in a System-on-Chip (SoC). TOE Physical scope also includes STT-MRAM memory macro from GlobalFoundries.

This Security Target claims conformance to the Secure Sub-System in System-on-Chip (3S in SoC) Protection Profile (**BSI-DSZ-CC-PP0117-2022**) and includes claims from JIL supporting document **Security Requirements for Post-Delivery Code Loading**.

2.4. TOE Overview

The Arm® CI-300P is a security enclave to be integrated into a SoC. It provides an isolated execution environment for executing sensitive processes and handling sensitive data. The security enclave is a Hard Macro synthesized independently of other components of the SoC in combination with a Hard Macro for MRAM.

Arm CI-300P can be used for various applications, where devices operate in a hostile environment and require a high-level of security, such as cellular subscriber identification, Smart Cards, user credentials storage, and payment. Its main usage is for Integrated SIMs.

The TOE contains a high-level of protection against tampering, fault injection, and side channel attacks, as well as others. The TOE uses a defense-in-depth approach, with process-dependent protection, digital and software mitigations.

The TOE allows dedicated applications such as SIM-OS to execute on the platform. Typically, the application would link with the Runtime Library provided by Arm and use cryptographic and security services provided by the platform.

The TOE allows execution of an application running from memory outside the boundaries of the TOE by providing security mechanisms that protect storage and execution of data and application in MRAM. To allow TOE security features, the MRAM should contain an NVR sector fully dedicated to the TOE. NVR is a memory sector controlled by dedicated hardware signals and exempt from chip erase.

2.4.1. Main security features of the TOE

CI-300P is an isolated execution environment with its own CPU and memories subsystem. It manages its own life-cycle state that determines a security policy. Lifecycle management allows disabling test and debug features when the device is deployed.

CI-300P can run an application from Non-secure external non-volatile memory (MRAM). Running such an application is protected by MSU, which is a secured, encrypted, and authenticated cache.

To protect application data residing in the external Non-secure memory against rollback, CI-300P provides a non-volatile monotonic counter. This counter resides in the NVR sector of the MRAM.

CI-300P provides hardware-supported cryptographic services with fault-injection and side-channel protection:

- Random number generation
- AES
  - Key generation and destruction
  - Encryption and decryption
  - Authentication
- ECC
  - Key generation and destruction
  - Signature generation and verification
  - Raw key agreement
- Hash
- Key derivation

All cryptographic services are available via a Secure cryptographic library.

CI-300P is equipped with physical protection against tampering, perturbation, and side-channel leakage, including (among others):

- An active shield that protects against physical probing and provides side-channel protection by generating electro-magnetic noise.
- Environmental sensors for checking operating conditions.
- Dedicated sensors for light detection.
- Reset tree protection.

CI-300P design contains parity protection for data processed by the processor, memories, and interconnects. Critical registers are duplicated.

To protect against side-channel attacks:

- Internal RAMs (TRAM and PKA RAM) are encrypted
- TRAM is address scrambled.
- Cryptographic engines are protected by digital means (for example, masking), as well as with Secure Power Convertor.

Attacks detected by any of these means are managed by an Alarm Management System, which routes the alarms to pre-defined safe responses aimed to prevent system operation under attacks.

To verify that only a Secure application runs on the device, CI-300P provides Secure boot services achieved through a series of bootloaders (FSB, SSB). In addition, the device allows Secure firmware update in the field.

Access to various memory regions is controlled through MPU. In general, runtime firmware running on the device is considered Secure, while one internal TOE asset (device unique identity, HUK) is not accessible to the runtime firmware.

2.5. TOE Description

2.5.1. Physical Scope of the TOE

CI-300P Hard Macro:

The Hard Macro is defined as layout in GDS-II format with related description files for SoC integration and verification.

If the SoC integrator does not have a Secure environment, the Hard Macro is delivered in two parts:

- The full layout file is delivered securely to a Silicon Foundry
- A Phantom-view of the layout is generated for SoC integration and verification and is delivered to SoC integrator.

If the SoC integrator has a Secure environment, the full Hard Macro layout file is delivered for SoC integration and verification.

The full layout GDS-II database of the Hard Macro includes FSB code in ROM.

MRAM Hard Macro:

An off-the-shelf MRAM memory instance provided by GlobalFoundries for their 22FDX technology.

Note

Although this Hard Macro is listed in the physical scope of the TOE, only the NVR sector is within TOE logical boundaries.

Software:

First Stage Bootloader (FSB)

ROM code implementing Secure boot (delivered with the Hard Macro).

Second Stage Bootloader (SSB)

Loaded to the OTP as part of the stages coming after silicon fabrications and before SoC delivery.

Firmware Update and Maintenance Tool (FUT)

Loaded to the MRAM as part of the stages coming after silicon fabrication and before SoC delivery.

Runtime Library

This library is linked into a non-TOE application, which is loaded to the device NVM (MRAM main array) at a stage later than OTP population (can be done for example at the device assembly facility). It can be loaded either before or after SoC delivery.

The following table defines all the deliverables and their versions.

Table 2-1: Deliverables and versions

Type	Item	Version	Details	Form of delivery	Method of delivery
Hard Macro	CI-300P Hard Macro	Layout marking: cl145:r0p0 Digital identification: PID: D2 B0 0b 04 CID: 0D F0 05 B1	Full layout in GDS-II format with related description files for SoC integration and verification, includes hardware design and FSB code in ROM	GDS-II	Digital delivery via FTP and IP-Merge internal to GlobalFoundries, or via Arm Connect, or via Arm Dropzone
	MRAM Hard Macro	MRAM_eFlash_128Kx144_2019q4v1 or MRAM_eFlash_384Kx144_2019q4v1	STT-MRAM memory macro from GlobalFoundries: MRAM_eFlash_128Kx144 or MRAM_eFlash_384Kx144 for 22FDX with internal ECC. Only NVR sector of this macro is considered within logical boundaries of the TOE. The delivery of the MRAM Hard macro is optional, whereas the IP-Merge of the MRAM hard macro is mandatory. Typically, MRAM Hard Macro is not delivered, the IP-Merge happens internally to GlobalFoundries	GDS-II	IP-Merge internal to GlobalFoundries

Type	Item	Version	Details	Form of delivery	Method of delivery
Layout phantom-view	CI-300P Hard Macro phantom view	cl145:r0p0	Phantom-view of the layout for SoC integration and verification. Phantom-view of the layout does not include RTL-based secrets nor ROM code. Design layers and hierarchy that can help identify digital standard cells and rebuild a netlist are removed. This delivery is optional and is only needed if SoC developer does not have a Secure room, see Chapter 2.6.	GDS-II	Arm digital delivery system
	MRAM Hard Macro phantom view	MRAM_eFlash_128Kx144_PA_2019q4v1 or MRAM_eFlash_384Kx144_PA_2019q4v1	Phantom-view of the layout of MRAM Hard Macro used for SoC implementation and substituted at IP-Merge	GDS-II	Digital delivery GlobalFoundries FoundryView
Firmware	FSB	1.1.1.0	First Stage Boot	Included in Hard Macro ROM	Part of Hard Macro delivery
	SSB	1.1.1.4	Second Stage Boot, to be provisioned into OTP	Software image	Arm digital delivery system
	FUT	1.3	Firmware Update and Maintenance Tool, to be provisioned into MRAM	Software image	Arm digital delivery system
Software	Runtime Library	1.1.5_1.0_1.0.6	Linkable library for integration into Runtime Firmware	Software image	Arm digital delivery system
Manufacturing Tools	CryptoIsland Provisioning Tools	r0p0-01eac0	A collection of tools needed for provisioning at manufacturing stage	Source code and software image	Arm digital delivery system

Type	Item	Version	Details	Form of delivery	Method of delivery
	CryptoIsland HW Testing Tools	r0p0-01eac0	A collection of tools needed for hardware testing at manufacturing stage	Source code and software image	Arm digital delivery system
	CryptoIsland Disablement Tool	r0p0-00eac0	A sample source for CI-300P disablement in case it is not desired to be used	Source code	Arm digital delivery system
	CryptoIsland Characterization Tools	r0p0-01eac0	A collection of tools needed for TRNG characterization at pre-manufacturing stage	Source code and software image	Arm digital delivery system
Guidance documentation	Arm® CryptoIsland ™-300P Operational Guidance	0000-06	Operational Guidance aimed for development of Embedded Software on CI-300P.	PDF	Arm digital delivery system
	Arm® CryptoIsland ™-300P Software Developers Manual	0000-03	Software API guide aimed for development of Embedded Software on CI-300P.	HTML	Arm digital delivery system
	Arm® CryptoIsland ™-300P Preparative Guidance	0000-04	Preparative Guidance describing one-time preparation of samples that have been provisioned at OSAT	PDF	Arm digital delivery system
	Arm® CryptoIsland ™-300P Software Release Note	4.0	Software release note document, includes information on software product acceptance, getting started and changes and changes from previous release	PDF	Arm digital delivery system
	Arm® CryptoIsland ™-300P Technical Reference Manual	0000-06	The Technical Reference Manual (TRM) describes the functionality of CryptoIsland-300P.	PDF	Arm digital delivery system

Type	Item	Version	Details	Form of delivery	Method of delivery
	Arm® CryptoIsland™-300P Hard Macro Implementation and Integration Guide	0001-06	The Implementation and Integration Guide provides the information that is required to integrate the Hard Macro into the So C.	PDF	Arm digital delivery system
	Arm® CryptoIsland™-300P Hard Macro Production Test Guide	0001-05	Description of the Design for Test capabilities that are provided with the product and how to use them (ABIST, LBIST, MBIST and OTP tests)	PDF	Arm digital delivery system
	Arm® CryptoIsland™-300P Hard Macro Qualification and Performance Report	02	Timing and power characteristics achieved in the sign-off simulation during digital implementation.	PDF	Arm digital delivery system
	Arm® CryptoIsland™-300P Manufacturing Guide	0000-02	Guidance for testing and provisioning of CI-300P at OSAT	PDF	Arm digital delivery system
	Arm® CryptoIsland™-300P Hard Macro Release Note	1.0	Hard Macro release note document, includes information on product acceptance, getting started and changes from previous release	PDF	Arm digital delivery system
	Arm® CryptoIsland™-300P Hard Macro Errata	1.0	Product Errata Notice	PDF	Arm digital delivery system

2.5.2. Logical scope of the TOE

The hardware part of the TOE consists of the following:

- A Secure processor
- Cryptographic engines with life-cycle state management
- Internal RAMs (Trusted RAM, LLRAM, PKA RAM)
- OTP NVM
- A module for communication with the SoC (MHU)
- Address Translation Unit (ATU)
- Secured Cache (MSU)
- An extension to monitor the interface to an MRAM dedicated sector (NVR), called NVM Gateway
- Secure Power Domain for cryptography side-channel protection
- Sensors protecting against perturbation and tampering

The software part of the TOE consists of a bootloader with two stages, Runtime Library, and a Firmware Update Tool (FUT).

2.5.3. Hardware features of the CI-300P Hard Macro

CPU

ARM V6-M Processor with Tamper Resistance

Cryptographic engines

- True random number generator (TRNG)
- AES cryptographic system equipped with countermeasures against different attacks
- PKA: asymmetric cryptographic system equipped with countermeasures against different attacks
- Hash cryptographic system

Memories

- ROM
- RAMs (TRAM, LLRAM, PKA RAM)
- L1 Instruction cache
- L2-like Secure instruction cache (MSU)
- Extension to MRAM (embedded MRAM) at the system level of the SoC (NVM Gateway) and NVR sector of MRAM
- Non-volatile, One Time Programmable (OTP) memory for credentials and code

Security peripherals and features

- Memory Security Unit (MSU) (Secure code execution from Non-secure remote memory)
- Tampering Control Unit (TCU)
- Key Management Unit (KMU)
- Physical security
  - Environmental Sensors (voltage, frequency, light, clock monitoring, reset tree monitoring)
  - Honey Pot network
  - Active Shield
  - Secure Power Converter
- TRAM (internal RAM) and PKA RAM encryption
- TRAM address scrambling
- Integrity protection for data at rest
- Integrity protection for data in transit

Interfaces

- Message handling Unit (MHU)
- Platform Signal Interface (PSI)
- Memory extension
- GPIO

2.5.4. TOE software

Bootloader divided into two stages

- First Stage Bootloader (FSB)
- Second Stage Bootloader (SSB)

Code loading utilities

- Firmware Update Tool (FUT)

Runtime Library that provides the following features:

Cryptographic services

- TRNG, DRBG
- AES functionality (key, generation, encryption, decryption, MAC)
- SHA-1, SHA-256
- ECC functionality (key generation, key exchange, signature, verification)
- KDF (NIST 800-108 CTR mode based on AES-CMAC)

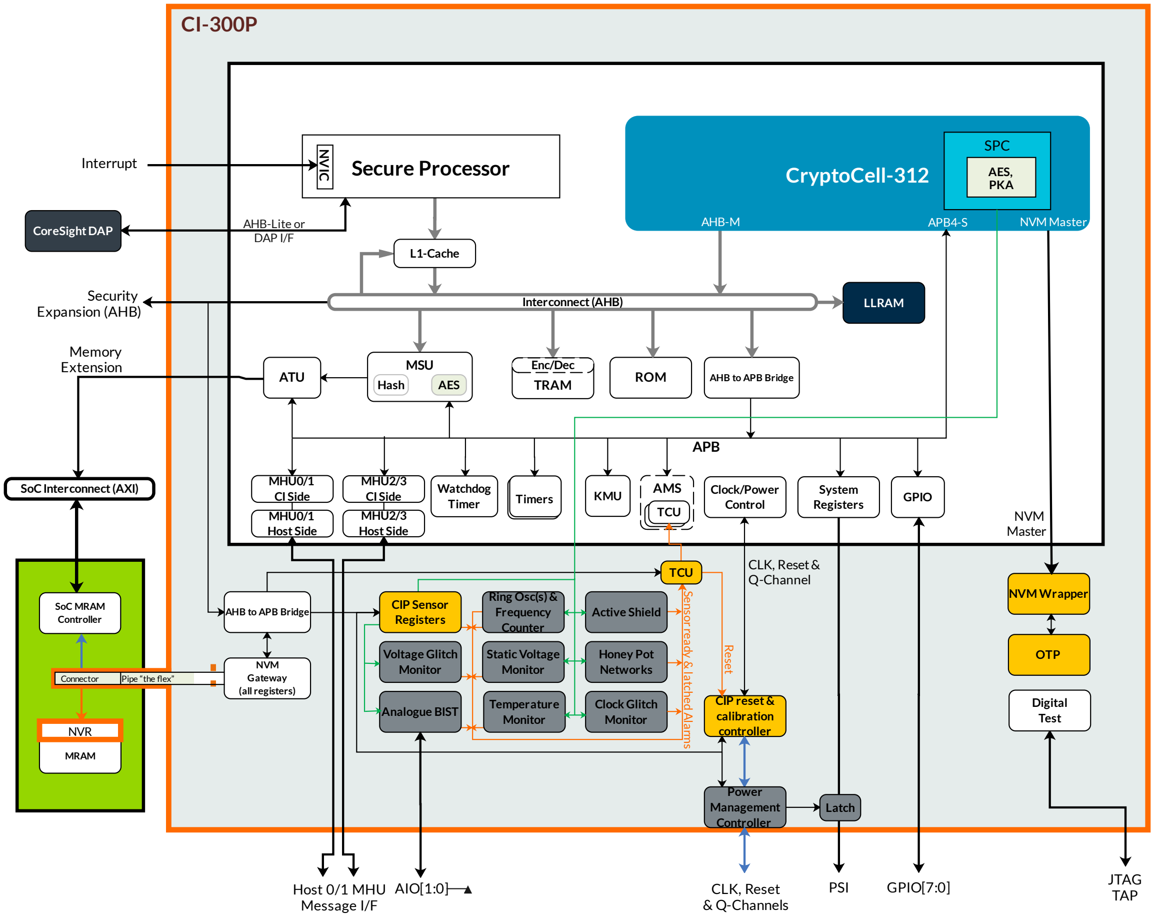
Additional services

- Non-volatile monotonic counter
- Hardware drivers: timers, watchdog, NVR access, ATU
- Diagnostics (BISTs)
- Firmware update
- RMA state transition
- Instance ID calculation and reporting.
- Components' version reporting
- Power management

2.5.5. TOE boundaries

CI-300P is an independent subsystem that operates in a larger SoC.

Figure 2-1: Functional diagram of the TOE



In Figure 2-1, the bold red line indicates the boundary of the TOE.

CI-300P uses MRAM memory that is shared with the rest of the SoC. The MRAM instance is structured into the main memory array, and Non-Volatile Registers (NVR).

Although the MRAM Hard Macro is listed in the physical scope of the TOE, the main array of the MRAM is outside the logical boundaries of the TOE, while NVR is considered within those boundaries. The MRAM controller resides outside of the TOE.

To secure the critical TOE data section of the (TOE-external) MRAM, an NVM Gateway is used, and is included in this TOE. The NVM Gateway monitors and controls NVR. This sector is exempt from chip erase and has dedicated access management signals.

Table 2-2: Primary TOE interfaces

#	Function	Description	Compliance
1	Memory Extension	CI-300P can access the memory location of the SoC, including SRAM and parts of MRAM. These memory locations are outside the TOE and are considered Non-secure.	AMBA® AHB5
2	Downstream Message (MHU Host Sender)	MHU is a mailbox-like interface used by CI-300P software to communicate with software running on SoC CPU.	AMBA APB3
3	Upstream Message (MHU Host Receiver)		
4	NVM Gateway	NVM Gateway is a hardware module (part of the TOE) aimed to secure the critical TOE data section in NVR sector.	AMBA APB3
5	Security Expansion (optional)	Optional. Can be used for adding control interface for CI-300P at SoC level.	AMBA AHB5
6	Debug	Debug interface, locked in deployed state of the TOE.	AMBA / Proprietary
7	GPIO	8 GPIOs for various indications from the SoC to the TOE.	N/A
8	Persistent State Interface (PSI)	PSI is a 16-bit interface containing critical indications about the TOE state intended for the SoC.	N/A
9a	Power Control	The power control interface.	AMBA Low Power Interface
9b	Clock Control	The clock control interface.	AMBA Low Power Interface
9c	Resets and Clocks	Reset and clock signals	N/A
10	Digital Test	Digital test interface used during CI-300P manufacturing. These interfaces are disabled when CI-300P is in deployed state.	IEEE 1149 TAP IEEE 1687 IJTAG
11	Analogue Test (optional)	Optional analogue test interface used during CI-300P manufacturing. These interfaces are disabled when CI-300P is in deployed state.	N/A

2.6. TOE Life Cycle

The TOE follows an amended version of the PP0117 Life Cycle flow, as described in Table 2-3. The role of the TOE Manufacturer during phase 2 (3S hardware development and integration into SoC) is split into activities performed by the Hard Macro Developer and activities performed by the SoC developer and identified as phases 2a and 2b.

The Hard Macro developer is responsible for the development of the TOE deliverables defined in Physical Scope of the TOE. The SoC developer is responsible for the development of the SoC that embeds the Security Hard Macro following guidance provided by the Hard Macro developer.

The TOE is delivered at the end of phase 2a for SoC integration.

The integration of the TOE in phase 2b uses one of the following:

- The full CI-300P Hard Macro if the SoC developer's environment is certified as Secure
- The phantom CI-300P Hard Macro view if the SoC developer does not have a Secure facility.

Typically, the phantom view of MRAM Hard Macro is used, unless agreed otherwise with GlobalFoundries. This agreement is contractual between GlobalFoundries and the SoC developer and is not related to whether the SoC developer has a Secure facility.

If the phantom view is used for SoC integration, substitution with the full CI-300P Hard Macro takes place during the data preparation steps for mask-making at the Silicon Foundry during phase 3. The same substitution process is done with the full MRAM Hard Macro.

Phases 4a and 4b are performed at an Outsourced Semiconductor Assembly and Test (OSAT) facility on behalf of the SoC developer. Depending on the type of assembly and particularly for wafer-level packages, the Silicon Foundry will provide partial layout information to the OSAT to enable lithography. This partial layout information does not include design layers used within the Security Hard Macro.

The SoC with integrated TOE is delivered at the end of phase 4b when the TOE state advances to Deployed. It is delivered to Composite Software Developer (a.k.a. Runtime Firmware developer) for application development on CI-300P, and to SoC developer for integration of the SoC with CI-300P.

Phase 6 is optional to allow personalization to be completed at the device level, separate from the provisioning step of phase 4b.

With the Security Hard Macro being one system within a complex SoC, phase 8 allows diagnostic investigations on parts returned from the field. The transition to this state involves the removal of all secrets from the TOE.

Table 2-3: TOE life cycle phases

Phase	Name	Location/owner	Details
1	3S Firmware and Software development	Composite Software Developer, Hard Macro developer	Development of embedded software
2a	3S hardware development	Hard Macro developer	Development of the Security Hard Macro.
TOE delivery			
2b	SoC Development and 3S integration	SoC developer	Development of the host SoC and TOE integration
3	3S in SoC Manufacturing	Silicon Foundry, Mask Shop	Phantom substitution (optional), mask generation and IC processing
4a	3S in SoC Packaging	OSAT	SoC assembly and test
4b	3S in SoC provisioning	OSAT	TOE is provisioned with SSB, FUT, and initialization data.
SoC including the 3S delivery			
5	3S in SoC Integration in PCB	Module or board manufacturer	SoC integration (board or module assembly)
6	3S in SoC Personalization	Personalize	SIM personalization. This stage is optional, SIM personalization can happen at stage 4b
7	3S in SoC Operation	Consumer of Composite Product (End-consumer)	The end user uses the devices including 3S in SoC. This may include loading applications in the field
8	Diagnostics, RMA	SoC Developer	Transition to this stage is done using a Secure procedure. All secrets and personalization data are deleted

3. Conformance Claims

3.1. CC Conformance Claim

This Security Target and the TOE claims to be conformant with version 3.1 of Common Criteria for Information Technology Security Evaluation according to:

- "Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017, CCMB-2017-04-001"
- "Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-002"
- "Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-003"

The following methodology is used for the evaluation:

- "Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 5, April 2017, CCMB-2017-04-004"

This Security Target and the TOE claims to be CC Part 2 extended and CC Part 3 conformant.

3.2. PP Claim

This Security Target claims strict conformance to the Protection Profile (PP) "Secure Sub-System in System-on-Chip (3S in SoC) Protection Profile", Version 1.5, registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-DSZ-CC-PP0117-2022.

BSI-DSZ-CC-PP0117-2022 claims strict conformance to the Protection Profile (PP) "Security IC Platform Protection Profile with Augmentation Packages", Version 1.0, registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0084-2014.

This Security Target does not claim conformance to any other Protection Profile.

3.3. Package Claim

The assurance level for this Security Target is EAL4+ augmented with ALC_DVS.2, ATE_DPT.2, ALC_FLR.2, and AVA_VAN.5.

The Security Target includes packages from the Protection Profile BSI-PP-0084-2014 and claims conformance as follows:

- Package "AES"
- Package "Hash-functions"
- Package "Loader dedicated for usage in secured environment only"

The Security Target includes packages from the Protection Profile BSI-DSZ-CC-PP0117-2022 and claims conformance as follows:

- Package for Passive External Memory

3.4. Conformance Claim Rationale

This security target claims strict conformance to a single PP, "Secure Sub-System in System-on-Chip (3S in SoC)", BSI-DSZ-CC-PP0117-2022.

Wherever the Protection Profile is mentioned in the ST, it always refers to Secure Sub-System in System-on-Chip (3S in SoC)", BSI-DSZ-CC-PP0117-2022, abbreviated PP-0117, unless explicitly indicated otherwise.

The Target of Evaluation (TOE) is a Hard Macro component to be integrated in an SoC.

The security problem definition of this security target is consistent with the statement of the security problem definition in the Protection Profile.

The security objectives of this Security Target are consistent with the statement of the security objectives in the Protection Profile.

The security requirements of this Security Target are consistent with the statement of the security requirements in the Protection Profile.

Additionally, the TOE aims at providing further cryptographic capacities to the users of the TOE.

The Organizational Security Policy P.Crypto-Service is refined to require the support of AES, Hash, KDF and ECC cryptographic functions.

The ST includes the following additional security objectives to enforce this refined Organizational Security Policy:

- O.AES
- O.SHA
- O.ECC
- O.KDF

The ST includes the following additional SFRs to meet these additional objectives:

- FCS_COP.1/AES
- FCS_CKM.1/AES
- FCS_COP.1/SHA
- FCS_COP.1/ECC
- FCS_CKM.1/ECC
- FCS_CKM.4/ECC
- FCS_CKM.1/KDF.

The TOE also adds Organizational Security Policy P.AdditionalCode-Loader for additional code loading after SoC delivery. It defines objectives related to security objectives for this process:

- O.Secure_Load_Acode
- O.Secure_AC_Activation
- O.TOE_Identification

Additional SFRs have been added to meet these objectives.

For protecting TOE data stored in remote memory, security objectives from the package "Passive External Memory Package" have been included, as follows:

- O.Pas-Mem-Cmd-Replay-Prot
- O.Pas-Mem-Unauth-Rollback-Prot
- O.Pas-Mem-Irreversible-Anchor
- O.Pas-Mem-Clone-Replace-Prot.

The following relevant SFRs have been added to meet these objectives:

- FPT_RPL.1/PM
- FDP_URC.1/PM
- FDP_IRA.1
- FDP_DAU.2/PM
- FIA_UID.1/PM
- FDP_IRA.1/PM is split into FDP_IRA.1/Data and FDP_IRA.1/Code to reflect that irreversibility anchors for data and code reside in different memory locations.
- FDP_SDC.1/PM is included in the general FDP_SDC.1
- FDP_SDI.1/PM is included in the general FDP_SDI.1

The A.Packaging-Requirement assumption and the corresponding OE.Packaging-Requirement do not apply to this TOE.

4. Security Problem Definition

4.1. Assets

The assets to be protected according to the section 3.1 of the Protection Profile, are

- user data of the TOE and the user data of the Composite Software
- TSF data, including root keys and keys derived from root keys, as well as the unique identification of the TOE instances
- firmware/software that is part of the TOE and the Composite Software, stored and in operation
- security services provided by the TOE for the Composite Software

The user (consumer) of the TOE places value upon the assets related to high-level security concerns:

SC1	integrity of user data of the Composite TOE
SC2	confidentiality of user data of the Composite TOE being stored in the TOE's protected memory areas.
SC3	correct operation of the security services provided by the TOE for the Composite Software
SC4	deficiency of random numbers

To be able to protect these assets (SC1 to SC4) the TOE shall self-protect its TSF.

Critical information about the TSF shall be protected by the development environment and the operational environment. Critical information may include physical design data, configuration data and layout data.

4.2. Threats

This section lists the threats that are defined in section 3.2 of the Protection Profile. They entirely apply to this Security Target.

Name	Description	Detailed description
T.Leak-Inherent	Inherent Information Leakage	An attacker may exploit information, as user data or TSF data, which is leaked from the TOE and/or the SoC interfaces while being stored and/or processed by the TOE
T.Phys-Probing	Physical Probing	An attacker may perform physical probing of the TOE. The probing is Performed • to disclose user data or TSF data while stored in protected memory areas, • to disclose/reconstruct user data or TSF data while processed or • to disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating user data of the composite TOE or the Composite Software.

Name	Description	Detailed description
T.Malfunction	Malfunction due to Environmental Stress	An attacker may cause a malfunction of TSF or of security services provided by the platform by applying environmental stress to the SoC or the 3S, to • modify security services of the TOE or • modify Composite Software including composite user data while being • processed by security services of the platform, or • deactivate or affect the TSF to enable disclosure or manipulation of user data. An attacker may also cause malfunction by • modifying data or messages, or by • misuse of architectural and micro architectural weaknesses via control and communication interfaces.
T.Phys-Manipulation	Physical Manipulation	An attacker may physically modify the TOE or the SoC, to • modify user data of the Composite Product, • modify the Composite Software, • modify or deactivate security services of the TOE, or • modify TSF of the TOE to enable attacks disclosing or manipulating TSF data, user data or the Composite Software
T.Leak-Forced	Forced Information Leakage	An attacker may disclose user data or TSF data, which is leaked from the TOE when such data is processed or stored by the TOE even if the information leakage is not inherent but caused by the attacker by influencing the TOE or the hosting SoC..
T.Abuse-Func	Abuse of Functionality	An attacker may misuse functions of the TOE which are disabled before the TOE is delivered. The misuse is applied, to • disclose or manipulate TSF data or user data, • manipulate (explore, bypass, deactivate or change) security services of the TOE or • manipulate (explore, bypass, deactivate or change) functions of the TOE FW/SW and of the Composite Software, or • enable an attack disclosing or manipulating user data or the Composite Software.
T.Insecure-State	Insecure State of the TOE	An attacker disturbs the boot process of the TOE by interrupting the boot process or introducing faults using T.Malfunction or T.PhysManipulation during start-up, which may force malicious code execution or TSF data manipulation. In this way, an attacker may • force invalid settings of the TOE hardware (e.g., life-cycle state, trimming, etc.), • load and execute unauthenticated firmware and/or software, • masquerade the unique identity, or • archive an inconsistent initialisation of the Root of Trust in order to • compromise secrets or enable other threats

Threats related to security services:

Name	Description	Detailed Description
T.RND	Deficiency of Random Numbers	An attacker manipulates or influences the random number generator to reduce the entropy, to predict or obtain information about random numbers generated by the TOE. An attacker may also predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided

The threat T.RND explicitly includes both deficiencies of hardware (true) random numbers as well as deficiency of software (pseudo) random numbers provided by the Crypto Library.

The following threats are related to protecting user data that resides in passive remote memory.

Name	Description	Detailed Description
T.Pas-Mem-Clone-Replace	Cloning or replacement of NVM	An attacker may attempt to clone the full content of the external memory or a specific memory area storing User Data of the 3S and write it to the external memory used by a different unit; alternatively, an attacker may physically replace the external memory used by a 3S with a different memory that may come from a different unit.
T.Pas-Mem-Content-Abuse	Abuse of passive external memory content	An attacker may attempt to access for disclosing or modifying the content of the external memory used by the 3S. Thereby an attacker may compromise confidentiality and/or integrity of TSF data and/or user data that shall be protected by the TOE
T.Pas-Mem-Cmd-Replay	Replay of commands between the 3S and the passive external memory	An attacker may attempt to replay the write and erase commands or responses to the read commands between the 3S and the passive external memory, to affect the freshness of the content read from or written to the external memory
T.Pas-Mem-Unauth-Rollback	Unauthorised rollback of content in the passive external memory	An attacker may attempt to read the content of the external memory, record it, and later write it back to the external memory after the original content were updated by the TOE.

4.3. Organizational Security Policies

The organizational security policies defined in section 3.3 of the Protection Profile PP-0117, and sections 7.3 and 7.4 of the PP-0084 apply to this Security Target.

Name	Description	Origin	Detailed Description
P.Gen-Unique-ID	Identification of each TOE instance	PP-0117	An accurate identification shall be established for the TOE. The policy requires that each instantiation of the TOE stores its own unique identification
P.Lim_Block_Loader	Limiting and Blocking the Loader Functionality	PP-0084	The composite manufacturer uses the Loader for loading of Composite Software, user data of the Composite Product or IC Dedicated Support Software in charge of the IC Manufacturer. He limits the capability and blocks the availability of the Loader in order to protect stored data from disclosure and manipulation.
P.Crypto-Service	Cryptographic services of the TOE	PP-0084	The TOE provides secure hardware based cryptographic services for the IC Embedded Software, as follows: • AES encryption and decryption, key generation • ECDSA signature generation and verification, ECC key generation, ECDH (ECC Diffie-Hellman) key exchange, key generation • SHA-1, SHA-224, SHA-256, DRBG • KDF
P.AdditionalCode-Loader	Additional code loading to the TOE	Security requirements for post-delivery code loading	The TOE provides a mechanism for secure loading of additional code into the initial TOE after the SoC delivery.

4.4. Assumptions

The following assumption is defined in section 3.4 of the Protection Profile.

Name	Description	Detailed Description
A.Resp-Appl	Treatment of user data of the Composite TOE	All user data of the Composite TOE are owned by Composite Software. Therefore, it must be assumed that security relevant user data of the Composite TOE (especially cryptographic keys) are treated by the Composite Software as defined for its specific application context.
A.Process-Sec-IC	Protection during Packaging, Finishing and Personalization	It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).

5. Security Objectives

5.1. Security Objectives for the TOE

The following standard high-level security goals are related to the assets:

SG1 Maintain the integrity of user data (when being executed/processed and when being stored in the TOE's memories).

SG2 Maintain the confidentiality of user data (when being processed and when being stored in the TOE's protected memories).

SG3 Maintain the correct operation of the security services provided by the TOE for the Composite Software.

SG4 Maintain the authenticity of the boot sequence and the setup of the root of trust

SG5 maintain the confidentiality, integrity and authenticity of the keys belonging to the Root of Trust

The integrity of TSF data as well as FW and SW as described in SG.1 are inherently covered because they are part of the TOE. Confidentiality is required for User Data. TSF data require confidentiality, in case the TSF data can be used to extract sensitive User Data without further information. The provisioning of random numbers is a security service covered by SG.3. The random numbers may also be used by the 3S, however, for internal purposes.

Note that the 3S does not distinguish between user data that are publicly known or kept confidential. Therefore, the 3S shall protect the user data in integrity and in confidentiality if stored in protected memory areas unless the Composite Software chooses to disclose or modify this user data.

These standard high-level security goals in the context of the security problem definition build the starting point for the definition of security objectives as required by the Common Criteria. Note that the integrity of the TOE is a means to reach these objectives.

5.1.1. Standard Security Objectives

Name	Description	Detailed Description
O.Leak-Inherent	Protection against Inherent Information Leakage	The TOE shall provide protection against disclosure of confidential TSF data and user data stored and/or processed in the 3S • by measurement and analysis of the shape and amplitude of any signal at the interfaces of the 3S (e.g., on the power, clock, or I/O lines) and/or • by measurement and analysis of the time between events found by measuring signals (e.g., on the power, clock, or I/O lines).
O.Phys-Probing	Protection against Physical Probing	The TOE shall provide protection against disclosure and reconstruction of user data or TSF data while stored in protected memory areas and processed by the TOE. This comprises also disclosure of other critical information about the operation of the TOE. This protection comprises • measuring through contacts which is direct physical probing on the chip surface except on pads being bonded (using standard tools for measuring voltage and current) or • measuring not using direct contacts but other types of physical interaction between charges (using tools used in solid-state physics research and IC failure analysis) with a prior reverse-engineering to understand the design and its properties and functions.
O.Malfunction	Protection against Malfunctions	The TOE must ensure its correct operation. The TOE must indicate or prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields
O.Phys-Manipulation	Protection against Physical Manipulation	The TOE must provide protection against manipulation of the TOE (including its software and TSF data), the Composite Software and the user data of the Composite TOE. This includes protection against: • reverse-engineering (understanding the design and its properties and functions), • manipulation of the hardware and any data, as well as • undetected manipulation of memory contents.
O.Leak-Forced	Protection against Forced Information Leakage	The TOE must be protected against disclosure of confidential data processed in the TOE (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker: • by forcing a malfunction (refer to "Protection against Malfunction due to Environmental Stress (O.Malfunction)" and/or • by a physical manipulation (refer to "Protection against Physical Manipulation (O.Phys-Manipulation)". If this is not the case, signals which normally do not contain significant information about secrets could become an information channel for a leakage attack.
O.Abuse-Func	Protection against Abuse of Functionality	The TOE must prevent that functions of the TOE which may not be used after TOE Delivery can be abused in order to (i) disclose critical user data of the Composite TOE, (ii) manipulate critical user data of the Composite TOE, (iii) manipulate Composite Software or (iv) bypass, deactivate, change or explore security features or security services of the TOE. Details depend, for instance, on the capabilities of the Test Features provided by the IC Dedicated Test Software that are not specified here.

Name	Description	Detailed Description
O.Secure-State	Secure start-up and re-start	The TOE shall be started through a secure initialisation process that Ensures • integrity and authenticity of code executed during startup, • integrity and authenticity of the hardware settings and the initialisation during start-up including the secure start-up of the Root of Trust functionality.
O.Identification	TOE Identification	The TOE must provide means to store Initialization Data and Pre-personalization Data in its non-volatile memory. The Initialization Data (or parts of them) are used for TOE identification.
O.RND	Random Numbers	The TOE will ensure the cryptographic quality of random number generation. For instance, random numbers shall not be predictable and shall have a sufficient entropy. The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys

5.1.2. Packages for Cryptographic Services

The TOE defines the following security objectives related to cryptographic services:

Name	Description	Origin	Detailed Description
O.AES	Cryptographic service AES	PP-0084	The TOE provides secure hardware based cryptographic services for the AES for encryption and decryption.
O.SHA	Cryptographic service Hash function	PP-0084	The TOE provides secure hardware based cryptographic services for secure hash calculation.
O.ECC	Cryptographic service ECC		The TOE provides secure ECC cryptographic services which are based on a combined hardware and software, for ECC sign and verify and key exchange. The TOE also provides ECC key pair generation.
O.KDF	Cryptographic service KDF		The TOE provides secure cryptographic services implementing the Key Derivation Function algorithm based on NIST SP 800-108 CTR mode.

5.1.3. Objectives for Loader

Name	Description	Origin	Detailed Description
O.Cap_Avail_Loader	Capability and availability of the Loader	PP-0084	The TSF provides limited capability of the Loader functionality and irreversible termination of the Loader to protect stored user data from disclosure and manipulation.

5.1.4. Objectives for Post Delivery Code Loader

Name	Description	Origin	Detailed Description
O. Secure_Load_ACode	Secure loading of the Additional Code	Security requirements for post-delivery code loading	The Loader of the Initial TOE shall check an evidence of authenticity and integrity of the loaded Additional Code. The Loader enforces that only the allowed version of the Additional Code can be loaded on the Initial TOE. The Loader shall forbid the loading of an Additional Code not intended to be assembled with the Initial TOE. During the Load Phase of an Additional Code, the TOE shall remain secure.
O.Secure_AC_Activation	Secure activation of the Additional Code	Security requirements for post-delivery code loading	Activation of the Additional Code and update of the Identification Data shall be performed at the same time in an Atomic way. All the operations needed for the code to be able to operate as in the Final TOE shall be completed before activation. If the Atomic Activation is successful, then the resulting product is the Final TOE, otherwise (in case of interruption or incident which prevents the forming of the Final TOE such as tearing, integrity violation, error case), the Initial TOE shall remain in its initial state or fail secure.
O.TOE_Identification	Secure identification of the TOE by the user	Security requirements for post-delivery code loading	The Identification Data identifies the Initial TOE and Additional Code. The TOE provides means to store Identification Data in its non-volatile memory and guarantees the integrity of these data. After Atomic Activation of the Additional Code, the Identification Data of the Final TOE allows identifications of Initial TOE and Additional Code. The user shall be able to uniquely identify Initial TOE and Additional Code(s) which are embedded in the Final TOE.

5.1.5. Security Objectives for Passive External Memory

Name	Description	Origin	Detailed Description
O.Pas-Mem-Content-Prot	Protection against disclosure and undetected modification of passive external memory content	PP-0117	The content in the external memory shall be protected against disclosure and undetected modification, because an attacker can directly access the external memory.
O.Pas-Mem-Cmd-Replay-Prot	Protection against replay of commands to store or modify data in passive external memory to the 3S	PP-0117	The TOE shall protect against replay of content during write, read and erase operations to the external memory by the 3S.
O.Pas-Mem-Unauth-Rollback-Prot	Protection against unauthorised rollback of external memory content	PP-0117	The TOE shall protect against replacement of the external memory content with a previous version, even if it was valid in the past
O.Pas-Mem-Irreversible-Anchor	Passive external memory content Irreversibility Anchor	PP-0117	The TOE shall implement a reference inside the 3S that represents the current content of the external memory. This reference shall be updated, based on each authorised modification of the external memory to ensure freshness of the data.

Name	Description	Origin	Detailed Description
O.Pas-Mem-Clone-Replace-Prot	Protection against passive external memory cloning or replacement	PP-0117	The TOE shall protect against cloning or replacement of user data with user data stored in the memory of another instance of the TOE and against replacement of the external memory with the one from another instance of the TOE

5.2. Security Objectives for the environment

5.2.1. Objectives for Loader

Name	Description	Detailed Description
OE.Lim_Block_Loader	Limitation of capability and blocking the Loader	The Composite Product Manufacturer will protect the Loader functionality against misuse, limit the capability of the Loader and terminate irreversibly the Loader after intended usage of the Loader.

5.2.2. Security Objectives for the Composite Software

The Composite Software shall provide "Treatment of user data of the Composite Product OE.RespAppl)", as specified below.

Name	Description	Detailed Description
OE.Resp-Appl	Treatment of user data of the Composite Product	Security relevant user data of the Composite Product (especially cryptographic keys) are treated by the Composite Software as required by the security needs of the specific application context

5.2.3. Security Objectives for Test and Pre-Personalisation of the 3S (Phases 3 to 5)

The pre-personalisation environment shall ensure "Uniqueness and authenticity of the device individual identifier" (OE.Secure-Initialisation).

Name	Description	Detailed Description
OE.Secure-Initialisation	Uniqueness and authenticity of the device individual identifier	Security procedures shall be applied during the initialisation of the TOE, to ensure that each device is loaded with an individual identifier. The identifier shall allow the unique identification of each device in later life cycle phases

5.2.4. Security Objectives for the Operational Environment after TOE Delivery

Appropriate “Protection during Packaging, Finishing and Personalisation (OE.Process-Sec-IC)” shall be ensured after TOE Delivery up to the end of Phase 5, as well as during the delivery to Phase 6 as specified below.

Name	Description	Detailed Description
OE.Process-Sec-IC	Protection during composite product manufacturing	Security procedures shall be used after TOE Delivery up to delivery to the end-user to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorized use).

5.3. Security Objectives Rationale

The table below gives an overview, how the assumptions, threats, and organizational security policies are addressed by the objectives.

Threat, organizational security policy or assumption	Security Objective	Justification
A.Resp-Appl	OE.Resp-Appl	OE.Resp-Appl requires the Composite Software to implement measures as assumed in A.Resp-Appl, so the assumption is covered by the objective.
A.Process-Sec-IC	OE.Process-Sec-IC	Since OE.Process-Sec-IC requires the Composite Product Manufacturer to implement those measures assumed in A.Process-Sec-IC, so the assumption is covered by this objective.
P.Gen-UniqueID	O.Identification OE.Secure-Initialisation	O.Identification requires that the TOE supports the possibility of a unique identification. The unique identification can be stored in the TOE. The unique identification is generated by the production environment, so the production environment shall support the integrity and initialisation of the generated unique identification as required by OE.Secure-Initialisation. The technical and organisational security measures that ensure the security of the testing and initialisation environment are evaluated, based on the assurance measures that are part of the evaluation. Therefore, the organisational security policy P.Gen-Unique-ID is covered by this objective, as far as organizational measures are concerned
T.Leak-Inherent	O.Leak-Inherent	For all threats the corresponding are stated in a way, which directly corresponds to the description of the objective. It is clear from the description of each objective, that the corresponding threat is removed if the objective is valid. More specifically, in every case the ability to use the attack method successfully is countered, if the objective holds.
T.Phys-Probing	O.Phys-Probing	
T.Malfunction	O.Malfunction	
T.Phys-Manipulation	O.Phys-Manipulation	
T.Abuse-Func	O.Abuse-Func	
T.RND	O.RND	

Threat, organizational security policy or assumption	Security Objective	Justification
T.Leak-Forced	O.Leak-Forced O.Malfunction O.Phys-Manipulation	T.Leak-Forced is countered by O.Leak-Forced, because the objective requires the protection against leakage even if the leakage is caused by an attacker trying to force malfunction and/or physical manipulation. Physical manipulation or environmental stress may be used to force leakage, so the protection against physical manipulation provided by O.Phys-Manipulation and the protection against malfunctions provided by O.Malfunction support the resistance against the threat T.Leak-Forced.
T.InsecureState	O.Secure-State	T.Insecure-State is countered by O.Secure-State, because the objective requires a secure initialization process that ensures integrity and authenticity of code executed during start-up as well as integrity and authenticity of the hardware configuration including the Root of Trust after start-up.
T.Pas-Mem-Content-Abuse	O.Pas-Mem-Content-Prot	T.Pas-Mem-Content-Abuse is countered by O.Pas-Mem-Content-Prot, which requires the TOE to prevent disclosure and undetected modification of the content stored in external memory
T.Pas-Mem-Cmd-Replay	O.NVM-Command-Replay-Protection O.Pas-Mem-Irreversible-Anchor	T.Pas-Mem-Cmd-Replay is countered by O.Pas-Mem-Cmd-Replay-Prot and O.Pas-Mem-Irreversible-Anchor as follows: O.Pas-Mem-Cmd-Replay-Prot requires protection against replay of commands exported from the 3S in the external memory mitigating T.Pas-Mem-Cmd-Replay. O.Pas-Mem-Irreversible-Anchor requires the implementation of a reference inside the 3S representing the current content of the external memory. The reference inside the 3S is updated associated with each change issued by the 3S on the external memory. This reference allows verification of the freshness of the data when they are loaded from the external memory.
T.Pas-Mem-Unauth-Rollback	O.Pas-Mem-Unauth-Rollback-Prot O.Pas-Mem-Irreversible-Anchor	T.Pas-Mem-Unauth-Rollback is countered by O.Pas-Mem-Unauth-Rollback-Prot and O.Pas-Mem-Irreversible-Anchor as follows: O.Pas-Mem-Unauth-Rollback-Prot requires that the TOE protects against replacement of external memory content with older content of the same external memory, where the data freshness property is not met, thereby mitigating this threat. O.Pas-Mem-Irreversible-Anchor requires that the TOE implements a reference inside the 3S representing the current content of the external memory. The reference inside the 3S is updated associated with each change issued by the 3S on the external memory. This reference allows verification of the freshness of the data when they are loaded from the external memory.
T.Pas-Mem-Clone-Replace	O.Pas-Mem-Clone-Replace-Prot	T.Pas-Mem-Clone-Replace is countered by O.Pas-Mem-Clone-Replace-Prot, which requires the TOE to detect the replacement of the external memory content with one of a different TOE's memory, or physical replacement of the external memory with the external memory of a different instance of the TOE.

Threat, organizational security policy or assumption	Security Objective	Justification
P.Crypto-Service	O.AES O.ECC O.SHA O.KDF	Since these objectives require the TOE to implement the same specific security functionality as required by P.Crypto-Service, the organization security policy is covered by the objective.
P.Lim_Block_Loader	O.Cap_Avail_Loader OE.Lim_Block_Loader	The organizational security policy "Limitation of capability and blocking the Loader" (P.Lim_Block_Loader) is directly implemented by the security objective for the TOE "Capability and availability of the Loader (O.Cap_Avail_Loader)" and the security objective for the TOE environment "Limitation of capability and blocking the Loader (OE.Lim_Block_Loader)". The TOE security objective "Capability and availability of the Loader" (O.Cap_Avail_Loader)" mitigates also the threat "Abuse of Functionality " (T.Abuse-Func) if attacker tries to misuse the Loader functionality in order to manipulate security services of the TOE provided or depending on IC Dedicated Support Software or user data of the TOE as IC Embedded Software, TSF data or user data of the smartcard product.
P.AdditionalCode-Loader	O.Secure_Load_ACode O.Secure_AC_Activation O.TOE_Identification	The organizational security "Post-delivery code loading to the TOE" (P.AdditionalCode-Loader) is directly implemented by the security objective for the TOE "Secure loading of the Additional Code (O.Secure_Load_Acode)", Secure activation of the Additional Code (O.Secure_AC_Activation) and Secure identification of the TOE by the user (O.TOE_Identification)

6. Extended Components Definition

This Security Target uses the extended security functional requirements defined in chapter 5 of the Protection Profile.

The following extended components as defined in the Protection Profile are used:

- FCS_RNG.1
- FMT_LIM.1
- FMT_LIM.2
- FAU_SAS.1
- FDP_SDC.1
- FPT_INI.1
- FDP_URC.1
- FDP_IRA.1

For the complete specification and justification of those extended SFRs, see Secure Sub-System in System-on-Chip Protection Profile.

7. Security Requirements

To define the Security Functional Requirements, Part 2 of the Common Criteria was used. However, some Security Functional Requirements have been refined. The refinements are described below the associated SFR.

The refinement operation is used to add detail to a requirement, and thus, further restricts a requirement. In such a case, an extra paragraph starting with "Refinement" may be given.

The selection operation is used to select one or more options provided by the CC in stating a requirement. Selections having been made by the ST author are denoted as ***bold and italicized***.

The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments having been made by the ST author appear in **bold text**. The iteration operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash "/", and the iteration indicator after the component identifier.

7.1. Security Functional Requirements (SFR)

7.1.1. Malfunctions

7.1.1.1. FRU_FLT.2/Env: Limited fault tolerance

FRU_FLT.2

Limited fault tolerance

Hierarchical to:

FRU_FLT.1 Degraded fault tolerance

Dependencies:

FPT_FLS.1 Failure with preservation of secure state

FRU_FLT.2.1/Env

The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: **exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1/Env)**

Refinement:

The term "failure" above means "circumstances". The TOE prevents failures for the "circumstances" defined above.

Application Note

Environmental conditions include but are not limited to power supply, clock, and other external signals (e.g., reset signal) necessary for the TOE operation.

7.1.1.2. FRU_FLT.2/Log: Limited fault tolerance

FRU_FLT.2

Limited fault tolerance

Hierarchical to:

FRU_FLT.1 Degraded fault tolerance

Dependencies:

FPT_FLS.1 Failure with preservation of secure state

FRU_FLT.2.1/Log

The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: abnormal interface behaviour and/or protocol parameters or protocol sequences that can be tolerated and that are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1/Log)

Refinement:

The term "failure" above means "circumstances". The TOE prevents failures for the "circumstances" defined above.

7.1.1.3. FPT_FLS.1/Env: Failure with preservation of secure state

FPT_FLS.1/Env

Failure with preservation of secure state

Hierarchical to:

No other components.

Dependencies:

No dependencies

FPT_FLS.1.1/Env

The TSF shall preserve a secure state when the following types of failures occur: exposure to operating conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2/Env) and where therefore a malfunction could occur.

Refinement:

The term "failure" above also covers "circumstances". The TOE prevents failures for the "circumstances" defined above.

7.1.1.4. FPT_FLS.1/Log: Failure with preservation of secure state

FPT_FLS.1/Log

Failure with preservation of secure state

Hierarchical to:

No other components.

Dependencies:

No dependencies

FPT_FLS.1.1/Log

The TSF shall preserve a secure state when the following types of failures occur: exposure to abnormal interface behaviour and/or protocol parameters or protocol sequences which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2/Log) and where, therefore, a malfunction could occur.

Refinement:

The term "failure" above also covers "circumstances". The TOE prevents failures for the "circumstances" defined above.

7.1.2. Abuse of functionality

7.1.2.1. FMT_LIM.1/Test: Limited Capabilities

FMT_LIM.1/Test

Limited capabilities

Hierarchical to:

No other components.

Dependencies:

FMT_LIM.2 Limited availability

FMT_LIM.1.1/Test

The TSF shall be designed in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks.

7.1.2.2. FMT_LIM.1/Debug: Limited Capabilities

FMT_LIM.1/Debug

Limited capabilities

Hierarchical to:

No other components.

Dependencies:

FMT_LIM.2 Limited availability

FMT_LIM.1.1/Debug

The TSF shall be designed in a manner that limits their capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: Deploying Debug Features after TOE Delivery does not allow user data of the Composite TOE to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks.

7.1.2.3. FMT_LIM.2/Test: Limited availability

FMT_LIM.2/Test

Limited availability

Hierarchical to:

No other components.

Dependencies:

FMT_LIM.1 Limited capabilities

FMT_LIM.2.1/Test

The TSF shall be designed in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: Deploying Test Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks.

7.1.2.4. FMT_LIM.2/Debug: Limited availability

FMT_LIM.2/Debug

Limited availability

Hierarchical to:

No other components.

Dependencies:

FMT_LIM.1 Limited capabilities

FMT_LIM.2.1/Debug

The TSF shall be designed in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: Deploying Debug Features after TOE Delivery does not allow User Data to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks.

7.1.3. Physical Manipulation and Probing

7.1.3.1. FDP_SDC.1: Stored data confidentiality

FDP_SDC.1

Storage data confidentiality

Hierarchical to:

No other components.

Dependencies:

No dependencies

FDP_SDC.1.1

The TSF shall ensure the confidentiality of the information of user data and dedicated TSF data while it is stored in the TRAM, OTP, PKA RAM, NVR partition of MRAM, and main MRAM array areas used to store confidential TOE data.

Application Note

FDP_SDC.1 covers both FDP_SDC.1/3S and FDP_SDC.1/PM defined in the Protection Profile.

7.1.3.2. FDP_SDI.2: Stored data integrity monitoring and action

FDP_SDI.2

Stored data integrity monitoring and action

Hierarchical to:

FDP_SDI.1 Stored data integrity monitoring

Dependencies:

No dependencies.

FDP_SDI.2.1

The TSF shall monitor user data stored in containers controlled by the TSF for **parity errors, ECC errors or cryptographic integrity errors** on all objects, based on the following attributes: **data stored in RAMs, sensitive data stored in OTP, sensitive peripherals registers, CPU registers, interconnect, NVR partition of MRAM, main MRAM array.**

FDP_SDI.2.2

Upon detection of a data integrity error, the TSF shall **enforce a reset or a non-maskable interrupt that eventually causes a reset, and increment an attacks counter stored in the NVR.**

Application Note

FDP_SDI.1 covers both FDP_SDI.1/3S and FDP_SDI.1/PM defined in the Protection Profile.

7.1.3.3. FPT_PHP.3: Resistance to physical attack

FPT_PHP.3

Resistance to physical attack

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_PHP.3.1

The TSF shall resist **physical manipulation and physical probing** to the **TSF** by responding automatically such that the SFRs are always enforced.

Refinement:

The TSF will implement appropriate mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TSF can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, automatic response means here

1. assuming that there might be an attack at any time and
2. countermeasures are provided at any time.

Application Note

This requirement is achieved by security feature as the shield must be removed and bypassed in order to perform physically intrusive attacks. The TOE executes an appropriate, secure reaction to stop operation if a physical manipulation or physical probing attack is detected. Additionally, internal scrambling & encryption for memories and logic area make the reverse-engineering of the TOE layout unpractical. These countermeasures combined meet the security functional requirement of FPT_PHP.3: Resistance to physical attack.

7.1.4. Leakage

7.1.4.1. FDP_ITT.1: Basic internal transfer protection

FDP_ITT.1

Basic internal transfer protection

Hierarchical to:

No other components.

Dependencies:

[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

FDP_ITT.1.1

The TSF shall enforce the Data Processing Policy to prevent the *disclosure* of user data when it is transmitted between physically-separated parts of the TOE.

Refinement:

The different memories, the CPU and other functional units of the TOE (e.g., a cryptographic co-processor) are seen as separate parts of the TOE.

Application Note

FDP_ITT.1/3S iteration identifier is not used because this ST does not claim Package for Secure External Memory.

7.1.4.2. FPT_ITT.1: Basic internal TSF data transfer protection

FPT_ITT.1

Basic internal transfer protection

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FPT_ITT.1.1

The TSF shall protect TSF data from *disclosure* when it is transmitted between separate parts of the TOE.

Refinement:

The different memories, the CPU and other functional units of the TOE (e.g., a cryptographic co-processor) are seen as separate parts of the TOE.

Application Note

FPT_ITT.1/3S iteration identifier is not used because this ST does not claim Package for Secure External Memory.

This requirement is equivalent to FDP_ITT.1 above but refers to TSF data instead of User Data. Therefore, it should be understood as to refer to the same Data Processing Policy defined under FDP_IFC.1 below.

7.1.4.3. FDP_IFC.1: Subset information flow control

FDP_IFC.1

Subset information flow control

Hierarchical to:

No other components.

Dependencies:

FDP_IFF.1 Simple security attributes

FDP_IFC.1.1

The TSF shall enforce the **Data Processing Policy** on **all confidential data when it is processed or transferred by the TOE or by the Composite Software.**

Application Note

The following Security Function Policy (SFP) Data Processing Policy is defined for the requirement "Subset information flow control (FDP_IFC.1)":

"User data and TSF data shall not be accessible from the TOE except when the firmware, software or Composite Software decides to communicate the user data of the Composite TOE via an external interface. The protection shall be applied to confidential data only but without the distinction of attributes controlled by the firmware, software and Composite Software."

Application Note

FDP_IFC.1/3S iteration identifier is not used because this ST does not claim Package for Secure External Memory.

7.1.5. TOE Identification and Root of Trust

7.1.5.1. FAU_SAS.1/Identification: Audit storage

FAU_SAS.1

Audit storage

Hierarchical to:

No other components.

Dependencies:

No dependencies

FAU_SAS.1.1

The TSF shall provide **the test process before TOE Delivery** with the capability to store ***the Initialization Data* and software components** in the **OTP and MRAM**.

Application Note

The integrity and uniqueness of the unique identification of the TOE must be supported by the development, production and test environment

Refinement:

The test process has been replaced with provisioning that runs after the integration of the Hard Macro and takes place after TOE Delivery.

7.1.5.2. FPT_INI.1: TSF Initialisation

FPT_INI.1

TSF Initialisation

Hierarchical to:

No other components.

Dependencies:

No dependencies

FPT_INI.1.1

The TOE initialization function shall **verify correct configuration of configurable and/or trimmable security mechanisms and the unique identification, integrity of start-up software, correct initialisation of internal keys**, **correct configuration of life cycle state,** prior to establishing the TSF in a secure initial state.

FPT_INI.1.2

The TOE initialization function shall detect and respond to errors and failures during initialization such that the TOE either successfully completes initialization or is halted.

FPT_INI.1.3

The TOE initialization function shall not be able to arbitrarily interact with the TSF after TOE initialization completes.

7.1.6. Random Numbers

7.1.6.1. FCS_RNG.1/TRNG: Random Number Generation

FCS_RNG.1

Random number generation

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FCS_RNG.1.1

The TSF shall provide a *physical* random number generator that implements:

- (PTG 2.1) A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.
- (PTG 2.2) If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source
- (PTG 2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.
- (PTG 2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.
- (PTG 2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered *continuously*. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.

FCS_RNG.1.2

The TSF shall provide *bits* that meet:

- (PTG 2.6) Test procedure A and NIST SP 800-22[7] tests suite does not distinguish the internal random numbers from output sequences of an ideal RNG.
- (PTG 2.7) The average Shannon entropy per internal random bit exceeds 0.99751.

Application Note

The random number generator is compliant to the definition of PTG.2 in AIS31[6].

7.1.6.2. FCS_RNG.1/DRBG: Random Number Generation

FCS_RNG.1

Random number generation

Hierarchical to:

No other components.

Dependencies:

No dependencies.

FCS_RNG.1.1

The TSF shall provide a *deterministic* random number generator that implements:

- (DRG.3.1) If initialized with a random seed using a PTRNG of class PTG.2 as random source, the internal state of the RNG shall have at least 128 bits of entropy.
- (DRG.3.2) The RNG provides forward secrecy.
- (DRG.3.3) The RNG provides backward secrecy even if the current internal state is known.

FCS_RNG.1.2

The TSF shall provide *random numbers* that meet:

- (DRG.3.4) The RNG, initialized with a random seed using a PTRNG of class PTG.2, generates output for which 2^35 strings of bit length 128 are mutually different with probability 1-2^17.
- (DRG.3.5) Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A and NIST SP 800-22[7] tests suite.

Application Note

The deterministic random number generator is compliant to the definition of DRG.3 in AIS31[6] and NIST 800-90A[5] section 10.2.

7.1.7. Packages for Cryptographic services

7.1.7.1. FCS_COP.1/AES: AES Operation

FCS_COP.1/AES

Cryptographic operation – AES

Hierarchical to:

No other components.

Dependencies:

[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/AES

The TSF shall perform decryption and encryption and authentication in accordance with a specified cryptographic algorithm AES in *ECB mode, CBC mode*, CTR mode, CMAC mode, CBC-MAC mode and cryptographic key sizes *128 bit, 192 bit, 256 bit* that meet the following: FIPS 197[1], NIST SP 800-38A[2], NIST SP 800-38B[3], ISO/IEC 9797-1[14].

7.1.7.2. FCS_COP.1/SHA: Hash Operation

FCS_COP.1/SHA

Cryptographic operation – SHA

Hierarchical to:

No other components.

Dependencies:

[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SHA

The TSF shall perform **hashing** in accordance with a specified cryptographic algorithm *SHA-1, SHA-224, SHA-256* and cryptographic key sizes **none** that meet the following **FIPS 180-4**[4].

Application Note

Hash is not considered a security related function and should not be used with secret values like keys, etc.

7.1.7.3. FCS_COP.1/ECC: ECC Operation

FCS_COP.1/ECC

ECC Cryptographic Operation

Hierarchical to:

No other components.

Dependencies:

[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/ECC

The TSF shall perform **signature generation, signature verification, Diffie-Hellman key agreement**, in accordance with a specified cryptographic algorithm **ECC over GF(p)** and cryptographic key sizes **256** that meet the following **FIP186-4** [11] **(ECDSA), NIST800-56A** [12] **(ECDH), SEC 1**[10].

Application Note

The following elliptic key curves are supported for signature generation, signature verification and Diffie-Hellman key agreement: NIST P-256 (as defined in SEC 2[8]), Brainpool P-256 (as defined in RFC5639[9]), FRP256v1 (as defined by ANSSI); the following curve for Diffie-Hellman key agreement only is supported: Curve25519 (as defined in RFC7748[15]).

7.1.8. Packages for Cryptographic functions

7.1.8.1. FCS_CKM.1/AES: AES Key Generation

FCS_CKM.1/AES

Cryptographic key generation - AES

Hierarchical to:

No other components.

Dependencies:

[FCS_CKM.2 Cryptographic key distribution, or

FCS_COP.1 Cryptographic operation]

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/AES

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm DRBG and specified cryptographic key sizes 128 bits, 192bits, 256 bits that meet the following NIST 800-90A[5] section 10.2.

7.1.8.2. FCS_CKM.4/AES: AES Key Destruction

FCS_CKM.4/AES

Cryptographic key destruction - AES

Hierarchical to:

No other components.

Dependencies:

[FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1/AES

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **overwriting the key with random string** that meets the following: none.

7.1.8.3. FCS_CKM.1/ECC: ECC Key Generation

FCS_CKM.1/ECC

Cryptographic key generation - ECC

Hierarchical to:

No other components.

Dependencies:

[FCS_CKM.2 Cryptographic key distribution, or

FCS_COP.1 Cryptographic operation]

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/ECC

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm ECC over GF(p) and specified cryptographic key sizes 256 that meet the following: NIST 800-56A[12].

Application Note

The following elliptic key curves are supported: NIST P-256, Brainpool P-256, FRP256v1 and Curve25519.

7.1.8.4. FCS_CKM.4/ECC: ECC Key Destruction

FCS_CKM.4/ECC

Cryptographic key destruction - ECC

Hierarchical to:

No other components.

Dependencies:

[FDP_ITC.1 Import of user data without security attributes, or

FDP_ITC.2 Import of user data with security attributes, or

FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1/ECC

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting the key with random string that meets the following: none.

7.1.8.5. FCS_CKM.1/KDF: Key Derivation

FCS_CKM.1/KDF

Key Derivation Function

Hierarchical to:

No other components.

Dependencies:

[FCS_CKM.2 Cryptographic key distribution, or

FCS_COP.1 Cryptographic operation]

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1/KDF

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm Key Derivation Function in Counter Mode using AES-CMAC and specified cryptographic key sizes 128 bits, 192bits, 256 bits that meet the following: NIST 800-108[13], NIST 800-38B[3], FIPS 197[1].

7.1.8.6. FCS_CKM.4/KDF: Key Destruction

FCS_CKM.4/KDF

Cryptographic key destruction - KDF

Hierarchical to:

No other components.

Dependencies:

[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1/KDF

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **overwriting the key with random string** that meets the following: **none**.

7.1.9. Packages for Loader

7.1.9.1. FMT_LIM.1/Loader: Loader Limited Capabilities

FMT_LIM.1

Loader Limited Capabilities

Hierarchical to:

No other components.

Dependencies:

FMT_LIM.2 Limited availability

FMT_LIM.1.1/Loader

The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with "Limited availability (FMT_LIM.2)" the following policy is enforced: **Deploying Loader functionality after moving to deployed state does not allow stored user data to be disclosed or manipulated by unauthorized user**

7.1.9.2. FMT_LIM.2/Loader: Loader Limited Availability

FMT_LIM.2

Loader Limited Availability

Hierarchical to:

No other components.

Dependencies:

FMT_LIM.1 Limited capabilities

FMT_LIM.2.1/Loader

The TSF shall be designed in a manner that limits its availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: **The TSF prevents deploying the Loader functionality after moving to deployed state.**

7.1.10. Post-delivery code and data loading

7.1.10.1. FDP_ITC.1: Import of user data without security attributes

FDP_ITC.1

Import of user data without security attributes

Hierarchical to:

No other components.

Dependencies:

[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FMT_MSA.3 Static attribute initialization

FDP_ITC.1.1

The TSF shall enforce the Firmware Update Policy when importing user data, controlled under the SFP, from outside of the TOE.

FDP_ITC.1.2

The TSF shall ignore any security attributes associated with the user data when imported from outside the TOE.

FDP_ITC.1.3

The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: none.

Application Note

The following Security Function Policy (SFP) Firmware Update Policy is defined for the requirement ": Import of user data without security attributes (FDP_ITC.1.1)":

"Upon firmware update, the TSF should: check integrity, authenticity and version validity of the candidate additional code, perform switching to the new code candidate in an atomic way, mark current firmware as invalid when starting firmware upgrade"

7.1.10.2. FAU_SAS.1/TOE_Identification: Audit storage

FAU_SAS.1

Audit storage

Hierarchical to:

No other components.

Dependencies:

No dependencies

FAU_SAS.1.1

The TSF shall provide the Firmware Update Tool with the capability to store the *Initialization Data* in the OTP and MRAM.

7.1.11. Remote memory protection

7.1.11.1. FDP_URC.1/PM: Protection against an unauthorised rollback of memory content

FDP_URC.1/PM

Protection against an unauthorised rollback of memory content

Hierarchical to:

No other components

Dependencies:

No dependencies

FDP_URC.1.1/PM

The TOE shall detect an unauthorised replacement of the content stored in passive external memory before the content is used. The detection shall be effective in any case where modification or read operation depends on the current content of this external memory.

FDP_URC.1.2/PM

Upon detection of unauthorised rollback of the content stored in a physically separated memory, the TOE shall *stop TOE operation*.

7.1.11.2. FDP_IRA.1/Data: Irreversibility Anchor for external memory

FDP_IRA.1/Data

Irreversibility Anchor for external memory

Hierarchical to:

No other components

Dependencies:

No dependencies

FDP_IRA.1.1/Data

The TSF shall verify the freshness of data for each read operation from the passive external memory.

FDP_IRA.1.2/Data

The Irreversibility Anchor shall maintain a distinct transaction reference for each *write, erase* operation and that is unambiguously linked with the current content of the transaction with the associated physically separated memory.

FDP_IRA.1.3/Data

The state of the Irreversibility Anchor implemented by the TSF shall be maintained during *any operation mode*.

Refinement:

The passive external memory is considered outside the TOE, even though it may be packaged together with the SoC including the 3S.

Application note

The Irreversibility Anchor is a monotonic counter stored in the NVR sector of the MRAM.

7.1.11.3. FDP_IRA.1/Code: Irreversibility Anchor for external memory

FDP_IRA.1/Code

Irreversibility Anchor for external memory

Hierarchical to:

No other components

Dependencies:

No dependencies

FDP_IRA.1.1/Code

The TSF shall verify the freshness of data for each read operation from the passive external memory.

FDP_IRA.1.2/Code

The Irreversibility Anchor shall maintain a distinct transaction reference for each *write, erase* operation and that is unambiguously linked with the current content of the transaction with the associated physically separated memory.

FDP_IRA.1.3/ Code

The state of the Irreversibility Anchor implemented by the TSF shall be maintained during *any operation mode*.

Refinement:

The 'data' for this SFR is restricted to code stored in the passive external memory.

Application note

The Irreversibility Anchor is a monotonic counter stored in the OTP.

7.1.11.4. FDP_DAU.2/PM: Data Authentication with Identity of Guarantor

FDP_DAU.2

Data Authentication with Identity of Guarantor

Hierarchical to:

FDP_DAU.1 Basic Data Authentication

Dependencies:

FIA_UID.1 Timing of identification

FDP_DAU.2.1/PM

The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of data objects and containers stored in the passive external memory.

FDP_DAU.2.2/PM

The TSF shall provide the 3S with the ability to verify evidence of the validity of the indicated information and the identity of the user that generated the evidence

Refinement:

The TSF generates the evidence that the data objects and containers stored in the external memory are generated by the dedicated 3S instance, based on FDP_IRA.1/Data, FDP_SDC.1 and FDP_SDI.2.

7.1.11.5. FIA_UID.1/PM: Timing of identification

FIA_UID.1

Timing of identification

Hierarchical to:

No other components

Dependencies:

No dependencies.

FIA_UID.1.1/PM

The TSF shall allow **any TSF-mediated actions that do not access data objects and/or containers stored in the external memory** on behalf of the user to be performed before the user is identified

FIA_UID.1.2/PM

The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Refinement:

The user is the 3S itself. The data objects and containers stored in the passive external memory need to be identified before any further action.

7.1.11.6. FPT_RPL.1/PM: Replay detection

FPT_RPL.1/PM

Replay detection

Hierarchical to:

No other components

Dependencies:

No dependencies

FPT_RPL.1.1/PM

The TSF shall detect replay for the following entities: **commands issued by the 3S to the passive external memory for the read, write and erase operations.**

FPT_RPL.1.2/PM

The TSF shall perform **a retry, then reset and recovery of the data using triple redundancy and increment attacks counter stored in the NVR when a replay is detected. If recovery is not possible, the TOE shall stop operation** when a replay is detected.

7.2. Security Assurance Requirements (SAR)

The security assurance requirements are as defined in the Protection Profile, including the refinements defined there.

7.3. Security Requirements Rationale

Security Objective	Security Functional Requirement
O.Leak-Inherent	**FDP_ITT.1: Basic internal transfer protection** **FPT_ITT.1: Basic internal TSF data transfer protection** **FDP_IFC.1: Subset information flow control**
O.Leak-Forced	**FDP_ITT.1: Basic internal transfer protection** **FPT_ITT.1: Basic internal TSF data transfer protection** **FDP_IFC.1: Subset information flow control** **FRU_FLT.2/Env: Limited fault tolerance** **FPT_FLS.1/Env: Failure with preservation of secure state** **FRU_FLT.2/Log: Limited fault tolerance** **FPT_FLS.1/Log Failure with preservation of secure state** **FPT_PHP.3: Resistance to physical attack**
O.Malfunction	**FRU_FLT.2/Env: Limited fault tolerance** **FPT_FLS.1/Env: Failure with preservation of secure state** **FRU_FLT.2/Log: Limited fault tolerance** **FPT_FLS.1/Log Failure with preservation of secure state** Supported by **FPT_INI.1 TSF Initialisation**
O.Phys-Probing	**FDP_SDC.1: Stored data confidentiality** **FPT_PHP.3: Resistance to physical attack**
O.Phys-Manipulation	**FDP_SDI.2: Stored data integrity monitoring and action** **FPT_PHP.3: Resistance to physical attack**
O.Abuse-Func	**FMT_LIM.1/Test: Limited Capabilities** **FMT_LIM.2/Test: Limited Availability** **FMT_LIM.1/Debug: Limited Capabilities** **FMT_LIM.2/Debug: Limited Availability** Supported by **FAU_SAS.1/Identification: Audit storage** **FRU_FLT.2/Env: Limited fault tolerance** **FPT_FLS.1/Env: Failure with preservation of secure state** **FRU_FLT.2/Log: Limited fault tolerance** **FPT_FLS.1/Log Failure with preservation of secure state** **FDP_SDI.2: Stored data integrity monitoring and action** **FPT_PHP.3: Resistance to physical attack**
O.Identification	**FAU_SAS.1/Identification: Audit storage** Supported by **FPT_INI.1 TSF Initialisation**

Security Objective	Security Functional Requirement
O.RND	**FCS_RNG.1/TRNG: Random Number Generation** **FCS_RNG.1/DRBG: Random Number Generation** Supported by **FRU_FLT.2/Env: Limited fault tolerance** **FPT_FLS.1/Env: Failure with preservation of secure state** **FDP_ITT.1: Basic internal transfer protection** **FPT_ITT.1: Basic internal TSF data transfer protection** **FDP_IFC.1: Subset information flow control** **FPT_PHP.3: Resistance to physical attack**
O.Secure-State	**FPT_INI.1 TSF Initialisation** Supported by **FRU_FLT.2/Env: Limited fault tolerance** **FPT_FLS.1/Env: Failure with preservation of secure state** **FRU_FLT.2/Log: Limited fault tolerance** **FPT_FLS.1/Log Failure with preservation of secure state** **FDP_SDI.2: Stored data integrity monitoring and action** **FPT_PHP.3: Resistance to physical attack**
O.AES	**FCS_COP.1/AES: AES Operation** **FCS_CKM.1/AES: AES Key Generation** **FCS_CKM.4/AES: AES Key Destruction**
O.SHA	**FCS_COP.1/SHA: Hash operation**
O.ECC	**FCS_COP.1/ECC: ECC operation** **FCS_CKM.1/ECC: ECC Key Generation** **FCS_CKM.4/ECC: ECC Key Destruction**
O.KDF	**FCS_CKM.1/KDF: Key Derivation** **FCS_CKM.4/KDF: Key Destruction**
O.Cap_Avail_Loader	**FMT_LIM.1/Loader: Loader Limited Capabilities** **FMT_LIM.2/Loader: Loader Limited Availability**
O.Secure_Load_ACode	**FDP_ITC.1: Import of user data without security attributes**
O.Secure_AC_Activation	**FDP_ITC.1: Import of user data without security attributes**
O.TOE_Identification	**FAU_SAS.1/TOE_Identification: Audit storage** Supported by **FPT_INI.1 TSF Initialisation**
O.Pas-Mem-Content-Prot	**FDP_SDC.1: Stored data confidentiality** **FDP_SDI.2: Stored data integrity monitoring and action**
O.Pas-Mem-Cmd-Replay-Prot	**FPT_RPL.1/PM: Replay detection**
O.Pas-Mem-Unauth-Rollback-Prot	**FDP_URC.1/PM: Protection against an unauthorised rollback of memory content** **FDP_IRA.1/Data: Irreversibility Anchor for external memory** **FDP_IRA.1/Code: Irreversibility Anchor for external memory**
O.Pas-Mem-Irreversible-Anchor	**FDP_IRA.1/Data: Irreversibility Anchor for external memory** **FDP_IRA.1/Code: Irreversibility Anchor for external memory**

Security Objective	Security Functional Requirement
O.Pas-Mem-Clone-Replace-Prot	**FDP_DAU.2/PM: Data Authentication with Identity of Guarantor** **FIA_UID.1/PM: Timing of identification**

7.3.1. O.Leak-Inherent

The refinements of the security functional requirements FPT_ITT.1 and FDP_ITT.1 together with the policy statement in FDP_IFC.1 explicitly require the prevention of disclosure of secret data (TSF data as well as user data) when transmitted between separate parts of the TOE or while being processed. This includes that attackers cannot reveal such data by measurements of emanations, power consumption or other behavior of the TOE while data are transmitted between or processed by TOE parts.

It is possible that the TOE needs additional support by the FW, SW and/or Composite Software (e.g., timing attacks are possible if the processing time of algorithms implemented in the software depends on the content of secret). This support must be addressed in the Guidance Documentation. Together with this FPT_ITT.1, FDP_ITT.1 and FDP_IFC.1 are suitable to meet the objective.

7.3.2. O.Leak-Forced

This objective is directed against attacks, where an attacker wants to force an information leakage, which would not occur under normal conditions. In order to achieve this the attacker has to combine a first attack step, which modifies the behavior of the TOE (either by exposing it to extreme operating conditions or by directly manipulating it) with a second attack step measuring and analyzing some output produced by the TOE. The first step is prevented by the same mechanisms which support O.Malfunction and O.Phys-Manipulation, respectively. The requirements covering O.Leak-Inherent also support O.Leak-Forced because they prevent the attacker from being successful if he tries the second step directly.

7.3.3. O.Malfunction

The definition of this objective covers situations where malfunction of the TOE might be caused by the operating conditions of the TOE or abnormal usage of TOE interfaces (while direct manipulation of the TOE is covered by O.Phys-Manipulation). For the operating conditions the security objective covers the following two circumstances: either all operating conditions are inside the tolerated range or at least one of them is outside this range. The second case is covered by FPT_FLS.1/Env, because it states that a secure state is preserved in this case. The first case is covered by FRU_FLT.2/Env, because it states that the TOE operates correctly under normal (tolerated) conditions. For the abnormal interface behaviour and/or protocol parameters or protocol sequences also two circumstances are covered:

Either the interface behaviour can be tolerated as described by FRU_FLT.2/Log or the interface behaviour may cause a mal function and, therefore, shall stop the operation and change to a secure state covered by FPT_FLS.1/Log. The TOE may enter the same a secure state for both iterations of FPT_FLS.1 or defines a secure state for each instance FPT_FLS.1/Env and FPT_FLS.1/Log.

The objective is supported by FPT_INI.1 that ensures the correct initialisation and configuration of the 3S during start-up. The functions implementing FRU_FLT.2/Env and FPT_FLS.1/Env shall work independently from the Composite Software so that their operation cannot be affected by the Composite Software. The functions implementing FRU_FLT.2/Log and FPT_FLS.1/Log shall apply for the interfacing between the TOE and the Composite Software as well as for the external interfaces provided by the TOE so that the different interfaces cannot be affected by the Security Services of the TOE. Therefore, there is no possible instance of conditions under O.Malfunction, which is not covered.

7.3.4. O.Phys-Probing

The SFR FDP_SDC.1 requires the TSF to protect the confidentiality of the information of the user data stored in specified memory areas and prevent its compromise by physical attacks bypassing the specified interfaces for memory access. The scenario of physical probing as described for this objective is explicitly included in the assignment chosen for the physical tampering scenarios in FPT_PHP.3. Therefore, it is clear that this security functional requirement supports the objective.

It is possible that the TOE needs additional support by the Composite Software (e.g., to send data over certain buses only with appropriate precautions). This support must be addressed in the Guidance Documentation. Together with this FPT_PHP.3 is suitable to meet the objective.

7.3.5. O.Phys-Manipulation

The SFR FDP_SDI.2 requires the TSF to detect the integrity errors of the stored user data and react in case of detected errors. The scenario of physical manipulation as described for this objective is explicitly included in the assignment chosen for the physical tampering scenarios in FPT_PHP.3. Therefore, it is clear that this security functional requirement supports the objective.

It is possible that the TOE needs additional support by the Embedded Software (for instance by implementing FDP_SDI.1 to check data integrity with the help of appropriate checksums, refer to Section 6.1). This support must be addressed in the Guidance Documentation. Together with this FPT_PHP.3 is suitable to meet the objective.

7.3.6. O.Abuse-Func

This objective states that abuse of test functions (especially provided by the firmware components that are used for product test, for example, to read data from memories) shall not be possible in Phase 5 of the life-cycle. There are two possibilities to achieve this:

(i) they cannot be used by an attacker (i.e., their availabilities are limited), or

(ii) using them would not provide an exploitable response for an attacker (i.e., their capabilities are limited) because the functions are designed in a specific way.

The limited availability is specified by FMT_LIM.2/Test and the limited capability is specified by FMT_LIM.1/Test. These requirements are combined to support the policy, which is suitable to fulfil O.Abuse-Func, so both SFRs together are suitable to meet the objective.

The two SFRs FMT_LIM.1/Debug and FMT_LIM.2/Debug are iterated, because debug functionality also needs to be disabled in Phase 5 of the life-cycle to prevent disclosure or modification of user data or TSF data using debug functionality. Debug functionality may be implemented with different security mechanisms to limit the capabilities and the availability of this functionality.

The SFR FAU_SAS.1 allows a unique identification of each TOE instance and thereby supports the protection against abuse. FRU_FLT.2/Env and FPT_FLS.1/Env control the operating conditions and prevent malfunctions that may allow to circumvent the control implemented by FMT_LIM.1 and FMT_LIM.2. FRU_FLT.1/Log and FPT_FLS.1/Log control the interface behaviour and prevent malfunctions that may allow to circumvent the control implemented by FMT_LIM.1 and FMT_LIM.2.

The SFR FDP_SDI.2 ensures the integrity of configuration data to ensure secure life-cycle control. The protection against manipulation as defined by FPT_PHP.3 prevents attackers from manipulation of the hardware.

FMT_LIM.1 and FMT_LIM.2 are explicitly (not using Part 2 of the Common Criteria) defined for the following reason: though taking components from the Common Criteria catalogue makes it easier to recognise functions, any selection from Part 2 of the Common Criteria would have made it harder for the reader to understand the special situation meant here. As a consequence, the statement of explicit SFRs was chosen to provide more clarity.

7.3.7. O.Identification

Obviously the operations for FAU_SAS.1 are chosen in a way that they require the TOE to provide the functionality needed for O.Identification. The Initialization Data (or parts of them) are used for TOE identification. The technical capability of the TOE to store Initialization Data and/or Pre-personalization Data is provided according to FAU_SAS.1.

It was chosen to define FAU_SAS.1 explicitly (not using a given security functional requirement from Part 2 of the Common Criteria) for the following reason: The security functional requirement FAU_GEN.1 in Part 2 of the CC requires the TOE to generate the audit data and gives details on the content of the audit records (for instance data and time). The possibility to use the functions in order to store security relevant data which are generated outside of the TOE, is not covered by the family FAU_GEN or by other families in Part 2. Moreover, the TOE cannot add time information to the records, because it has no real time clock. Therefore, the new family FAU_SAS was defined for this situation.

7.3.8. O.RND

FCS_RNG.1/TRNG requires the TOE hardware to provide random numbers of good quality. FCS_RNG.1/DRBG requires the TOE to provide random numbers of good quality sourced from TOE hardware. The exact metrics are specified within this Security Target.

Other security functional requirements, which prevent physical manipulation and malfunction of the TOE (see the corresponding objectives listed in the table) support this objective because they prevent attackers from manipulating or otherwise affecting the random number generator.

Random numbers are often used by the Composite Software to generate cryptographic keys for internal use. Therefore, the TOE must prevent the unauthorized disclosure of random numbers. Other security functional requirements which prevent inherent leakage attacks, probing and forced leakage attacks ensure the confidentiality of the random numbers provided by the TOE.

Depending on the functionality of specific TOEs the Composite Software will have to support the objective by providing runtime-tests of the random number generator. Together, these requirements allow the TOE to provide cryptographically good random numbers and to ensure that no information about the produced random numbers is available to an attacker.

It was chosen to define FCS_RNG.1 explicitly with FCS_RNG.1 /TRNG and FCS_RNG.1/DRBG iterations because Part 2 of the Common Criteria do not contain generic security functional requirements for Random Number generation. (Note, that there are security functional requirements in Part 2 of the Common Criteria, which refer to random numbers. However, they define requirements only for the authentication context, which is only one of the possible applications of random numbers.)

7.3.9. O.Secure-State

The SFR FPT_INI.1 implements security mechanisms to verify the correct configuration of the required parameter (e.g., trimming and life-cycle control) and the unique identification during the start-up. Further on, the SFR requires an integrity protection of the software executed during startup and the correct initialisation of internal keys as required by the objective. Therefore, FPT_INI.1 is suitable to meet the objective.
The security objective O.Secure-State is supported by FRU_FLT.2/Env and FPT_FLS.1/Env controlling the operating conditions and FRU_FLT.1/Log and FPT_FLS.1/Log controlling the interface behaviour prevent malfunctions that may allow to manipulate the secure initialisation. The SFR FDP_SDI.2 ensures the integrity of configuration data. The protection against manipulation as defined by FPT_PHP.3 prevents attackers from manipulation of the hardware to circumvent the secure initialisation.

7.3.10. O.AES

FCS_COP.1/AES, FCS_CKM.1/AES and FCS_CKM.4/AES meet the security objective 'Cryptographic service AES (O.AES)' as they require AES operation, AES key generation and AES key destruction.

7.3.11. O.SHA

FCS_COP.1.1/SHA meets the security objective ‘Cryptographic service SHA (O.SHA)’ as it requires SHA operation.

7.3.12. O.ECC

FCS_COP.1/ECC, FCS_CKM.1/ECC and FCS_CKM.4/ECC meet the security objective 'Cryptographic service ECC (O.ECC)' as they require ECC operation, ECC key generation and ECC key destruction.

7.3.13. O.KDF

FCS_CKM.1/KDF and FCS_CKM.4/KDF meet the security objective 'Cryptographic service ECC (O.ECC)' as they require key derivation and key destruction.

7.3.14. O.Cap_Avail_Loader

The security objective “Capability and availability of the Loader (O.Cap_Avail_Loader) is directly covered by the SFR FMT_LIM.1/Loader and FMT_LIM.2/Loader.

7.3.15. O.Secure_Load_ACode

FDP_ITC.1 requires to check integrity, authenticity and version validity of the candidate additional code, perform switching to the new code candidate in an atomic way, and mark current firmware as invalid when starting firmware upgrade. In this way, this security functional requirement meets the objective.

7.3.16. O.Secure_AC_Activation

FDP_ITC.1 requires among others that the TSF perform switching to the new code candidate in an atomic way. In this way this security functional requirement meets the objective.

7.3.17. O.TOE_Identification

The operations for FAU_SAS.1 are chosen in a way that they require the TOE to provide the Identification Data in its non-volatile memory. The Initialization Data of the Final TOE allows identifications of Initial TOE and Additional Code. The technical capability of the TOE to store Initialization Data and/or Pre-personalization Data is provided according to FAU_SAS.1.

7.3.18. O.Pas-Mem-Content-Prot

FDP_SDC.1 ensures protection of confidentiality of the contents stored in the external NVM, while the FDP_SDI.2 ensures protection of the integrity of the contents stored in the NVM. Therefore, it is clear that these security functional requirements support the objective.

7.3.19. O.Pas-Mem-Cmd-Replay-Prot

FPT_RPL.1 requires the TSF to detect replays in responses in the read commands to the NVM or replays of sequences of write/erase commands to the external NVM. This requirement is considered in the assignment of FPT_RPL.1.1. Therefore, it is clear that this security functional requirement supports the objective.

7.3.20. O.Pas-Mem-Unauth-Rollback-Prot

FDP_URC.1/PM requires that the TSF detects the case when the contents of the external NVM have been replaced by previous versions of them. This way, this security functional requirement supports the objective.

7.3.21. O.Pas-Mem-Irreversible-Anchor

FDP_IRA.1/Data and FDP_IRA.1/Code require the TOE to implement non-volatile mechanisms that are irreversible and serve to mark the NVM contents as meeting or not meeting data and code freshness property. By providing the mechanisms required by those SFRs, the security objective O.Pas-Mem-Irreversible-Anchor is directly supported.

7.3.22. O.Pas-Mem-Clone-Replace-Prot

The SFR FDP_DAU.2/PM requires the TOE to be able to generate evidence that guarantees the validity of data objects and containers stored in the external memory. The cloning or replacement of the external memory is detected, based on FIA_UID.1/PM, which requires the user identification before any data objects or containers stored in the external memory are accessed. By providing the mechanism required by these two SFRs, the security objective O.Pas-Mem-Clone-Replace-Prot is directly supported.

7.3.23. Dependencies of security functional requirements

Table 7-1: Security functional requirements dependencies

Security Functional Requirement	Dependencies	Fulfilled by security requirements
FRU_FLT.2/Env	FPT_FLS.1/Env	Yes
FPT_FLS.1/Env	None	No dependency
FRU_FLT.2/Log	FPT_FLS.1/Log	Yes
FPT_FLS.1/Log	None	No dependency
FMT_LIM.1/Test	FMT_LIM.2/Test	Yes
FMT_LIM.2/Test	FMT_LIM.1/Test	Yes
FMT_LIM.1/Debug	FMT_LIM.2/ Debug	Yes
FMT_LIM.2/ Debug	FMT_LIM.1/ Debug	Yes
FAU_SAS.1	None	No dependency
FDP_SDC.1	None	No dependency
FDP_SDI.2	None	No dependency
FPT_PHP.3	None	No dependency
FPT_ITT.1	None	Yes
FDP_ITT.1	FDP_ACC.1 or FDP_IFC.1	Yes
FDP_IFC.1	FDP_IFF.1	No, as explained in the PP, chapter 6.3.2
FCS_RNG.1/TRNG	None	No dependency
FCS_RNG.1/DRBG	None	No dependency
FCS_COP.1/AES	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FCS_CKM.1/AES FCS_CKM.4/AES

Security Functional Requirement	Dependencies	Fulfilled by security requirements
FCS_COP.1/SHA	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	Hash generation does not require a key, therefore key generation dependency is not fulfilled. It does not operate secret values, therefore key destruction dependency is not fulfilled
FCS_COP.1/ECC	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4	FCS_CKM.1/ECC FCS_CKM.4/ECC
FCS_CKM.1/AES	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1/AES FCS_CKM.4/AES
FCS_CKM.1/ECC	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_COP.1/ECC FCS_CKM.4/ECC
FCS_CKM.1/KDF	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4	FCS_CKM.4/KDF
FCS_CKM.4/AES	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_COP.1/AES FCS_CKM.1/AES
FCS_CKM.4/ECC	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_COP.1/ECC FCS_CKM.1/ECC
FCS_CKM.4/KDF	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	FCS_CKM.1/KDF
FMT_LIM.2/Loader	FMT_LIM.1/Loader	Yes
FMT_LIM.1/Loader	FMT_LIM.2/Loader	Yes
FPT_RPL.1/PM	None	No dependency
FDP_URC.1/PM	None	No dependency
FDP_IRA.1/Data	None	No dependency
FDP_IRA.1/Code	None	No dependency
FPT_INI.1	None	No dependency
FDP_ITC.1	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	No, the SFR is sufficiently described in Firmware Update Policy

7.4. Rationale for the Security Assurance Requirements

An assurance level of EAL4+ with the augmentations AVA_VAN.5, ALC_DVS.2 and ALC_FLR.2 are required for this type of TOE since it is intended to defend against sophisticated attacks. This evaluation assurance package was selected to permit a developer to gain maximum assurance from positive security engineering based on good commercial practices. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defense against such attacks, the evaluators should have access to the low-level design and source code.

7.4.1. AVA_VAN.5 Advanced methodical vulnerability analysis

Due to the intended use of the TOE, it must be shown to be highly resistant to penetration attacks. This assurance requirement is achieved by the AVA_VAN.5 component.

Independent vulnerability analysis is based on highly detailed technical information. The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing high attack potential.

AVA_VAN.5 has dependencies to ADV_ARC.1 "Security architecture description", ADV_FSP.2 "Security enforcing functional specification", ADV_TDS.3 "Basic modular design", ADV_IMP.1 "Implementation representation of the TSF", AGD_OPE.1 "Operational user guidance", and AGD_PRE.1 "Preparative procedures". All these dependencies are satisfied by EAL4.

It has to be assumed that attackers with high attack potential will try to attack secure elements used for digital signature applications or payment systems. Therefore, specifically AVA_VAN.5 was chosen to assure that even these attackers cannot successfully attack the TOE.

7.4.2. ALC_DVS.2 Sufficiency of security measures

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE.

This assurance component is a higher hierarchical component to EAL4 (which only requires ALC_DVS.1). ALC_DVS.2 has no dependencies.

7.4.3. ALC_FLR.2 Flaw reporting procedures

The augmentation with ALC_FLR.2 has been chosen to achieve a secure continuous operation of the TOE.

The flaw remediation process includes the possibility for users to report identify failures, flaws and abnormal behaviour to the developer. The developer needs an internal tracking and assessment of these issues. Furthermore the developer needs to implement corrective actions and deliver information on the flaw, corrections and guidance on corrective actions to TOE users. This provides assurance that the TOE will be maintained and supported in the future, requiring the TOE developer to track and correct flaws in the TOE.

ALC_FLR.2 has no dependencies.

ALC_FLR.2 is not included in the defined assurance level.

8. TOE Summary Specification

The TOE Summary Specification lists security services and security functions aimed to meet the security functional requirements. The TOE Summary Specification Rationale contains a map of the security services and countermeasures versus the met requirements.

8.1. Integrity protection (SF_INT)

The integrity of TOE internal RAM (TRAM and PKA RAM), critical registers, critical data in the OTP, data residing in the NVR and code residing in the external MRAM is protected by various mechanisms, which provide functionality required by FDP_SDI.2 "Stored data integrity monitoring and action".

8.2. Confidentiality protection (SF_CONF)

The confidentiality of data residing in NVR, code residing in external MRAM, contents of internal RAMs, critical data residing in the LLRAM is protected using various mechanisms, which provide functionality required by FDP_SDC.1 "Stored data confidentiality", FPT_ITT.1 "Basic internal TSF data transfer protection", FDP_ITT.1 "Basic internal transfer protection", FDP_IFC.1 "Subset information flow control".

8.3. Physical attacks protection (SF_PHYS)

8.3.1. Physical protection

The TOE is equipped with several physical protections against physical attacks, such as active shield, analogue sensors, digital sensors, as well as digital means such as integrity protection (SD_INT).

Software components of the TOE are programmed using software techniques that detect injected faults.

All the mechanisms above provide functionality for FPT_PHP.3 "Resistance to physical attack", FRU_FLT.2 "Limited fault tolerance", FPT_FLS.1 "Failure with preservation of secure state".

8.4. Side-channel protection (SF_SC)

8.4.1.1. TRAM and PKA RAM encryption and address scrambling

The TOE internal RAMs (TRAM and PKA RAM) are protected against side-channel leakage.

Cryptographic services and functions are implemented using side-channel protected cryptographic engines. The code of cryptographic services and functions as well the code that handles the secrets non-cryptographically is written in a way that prevents side-channel leakage.

All these mechanisms provide functionality required by FPT_ITT.1 “Basic internal TSF data transfer protection”, FDP_ITT.1 “Basic internal transfer protection”.

8.5. Access control (SF_AC)

The TOE provides access control for areas residing in remote memory and critical assets residing in the OTP. These mechanisms provide functionality required by FDP_IFC.1 “Subset information flow control”.

8.6. External memory protection (SF_EM)

8.6.1. Protection of code outside the TOE

The confidentiality, integrity, authenticity of code stored in the non-volatile memory outside the TOE boundary is ensured integrity protection and on-the-fly decryption. This mechanism provides functionality required by FDP_SDC.1 “Stored data confidentiality” and FDP_SDI.2 “Stored data integrity monitoring and action”.

Protection against NVM cloning or replacement is ensured by maintaining device-unique encryption and authentication codes for code residing in remote memory. This fulfils FDP_DAU.2/PM “Data Authentication with Identity of Guarantor”. The authentication codes are checked before running the decrypted code, which fulfils FIA_UID.1/PM “Timing of identification”.

Freshness of code residing in the remote memory (MRAM main array) is assured using a monotonic counter residing in the OTP memory. The FUT component responsible for post-delivery code update is also responsible for incrementing this counter. This fulfils FDP_IRA.1/Code “Irreversibility Anchor for external memory” and FDP_URC.1/PM “Protection against an unauthorised rollback of memory content”.

8.6.2. Protection of NVM data outside the TOE

The confidentiality, integrity, and authenticity of data stored in the non-volatile memory outside the TOE boundary is ensured by cryptographic mechanisms. It is the responsibility of the Composite Software to use cryptographic services provided by the TOE to ensure user data, as mentioned in Guidance Documentation (*Arm® CryptoIsland™-300P Operational Guidance*). Following this guidance fulfils FDP_DAU.2/PM "Data Authentication with Identity of Guarantor", FIA_UID.1/PM "Timing of identification", FDP_SDC.1 "Stored data confidentiality", FDP_SDI.2 "Stored data integrity monitoring and action", FDP_DAU.2/PM "Data Authentication with Identity of Guarantor" and FIA_UID.1/PM "Timing of identification".

To ensure freshness of user data, a monotonic counter is managed in the NVR sector of MRAM. This monotonic counter should be used by Composite Software to protect its data from replay and rollback by defining an encryption scheme that would involve a device-unique key and the counter. The contents of the NVR are protected by multiple security mechanisms. This fulfils FDP_IRA.1/Data "Irreversibility Anchor for external memory" and FDP_URC.1/PM "Protection against an unauthorised rollback of memory content".

8.7. Alarm management (SF_AM)

Alarm management mechanisms provide functionality required by FRU_FLT.2 "Limited fault tolerance" and FPT_FLS.1 "Failure with preservation of secure state".

8.8. Device unique Identity (SF_DUI)

The TOE maintains a device hardware unique key (HUK) that serves as a source for generating device unique identity. The HUK is stored in the OTP. It might be used for various device unique identifiers, both internal and external, for example attestation. It is generated during the TOE personalization stage, and it is not directly accessible to the Composite Software.

The device unique identity is used for an attestation service, which allows the Composite Software to attest its state. This provides functionality required by FAU_SAS.1 "Audit storage".

8.9. Life-cycle management (SF_LC)

The TOE implements life-cycle control to define security policies of the TOE functionality.

The life-cycle states supported by the TOE are as follows:

- Chip Manufacturing
- Device Manufacturing
- Deployed
- RMA

The mechanisms above provide functionality required by FMT_LIM.1/Test "Limited capabilities" and FMT_LIM.2/Test "Limited availability" as well as by FMT_LIM.1/Debug "Limited capabilities" and FMT_LIM.2/Debug "Limited availability".

8.10. Secure boot, firmware update (SF_BFU)

The TOE provides Secure boot process which is comprised of several stages. The Secure boot flow differs for various life-cycle states (see **Life cycle management (SF_LC)**) and supports firmware update in deployed life-cycle state.

The Secure boot mechanism provides functionality required by FPT_INI.1 "TSF Initialisation", FDP_URC.1/PM "Protection against an unauthorised rollback of memory content" by verifying firmware version freshness. It also provides functionality required by FDP_DAU.2/PM "Protection against an unauthorised rollback of memory content" by using MSU with device unique encryption and supports FMT_LIM.1/Loader "Limited capabilities" and FMT_LIM.2/Loader "Limited availability" by implementing different flows for Deployed and Chip Manufacturing life-cycle states.

The Firmware Update mechanism provides functionality required by FDP_ITC.1 "Import of user data without security attributes" by implementing Firmware Update policy.

8.11. Random numbers generation (SF_RNG)

The TOE implements physical hardware for generating a stream of random bits based on a random source clock – True Random Number Generator (TRNG). The stream of random bits can be used for generating seeds for Deterministic Random Bit Generation (DRBG). The TOE provides a DRBG driver for the Composite Software usage. This provides functionality required by FCS_RNG.1/TRNG and FCS_RNG.1/DRBG "Quality metric for random numbers".

8.12. AES (SF_AES)

The TOE implements an AES hardware engine. The engine supports operations with 128, 192 and 256 bit keys and implements the ECB, CBC, CTR encryption and decryption as well as CMAC and CBC-MAC authentication modes of operation. To operate it, the TOE provides an AES driver for Composite Software usage. The engine mitigates against Side Channel Analysis and Fault Injection attacks. The combination of engine and driver provide functionality required by FCS_COP.1/AES "Cryptographic operation – AES", FCS_CKM.1/AES "Cryptographic key generation – AES", and FCS_CKM.4/AES "Cryptographic key destruction - AES".

8.13. SHA (SF_SHA)

The TOE implements a SHA hardware engine. The engine supports the SHA-1, SHA-224 and SHA-256 algorithms. To operate it, the TOE provides a hash driver for Composite Software usage. The combination of engine and driver provides functionality required by FCS_COP.1/SHA "Cryptographic operation – SHA".

8.14. ECC (SF_ECC)

The TOE implements a PKA hardware engine to accelerate public key cryptography. In addition, the TOE provides software drivers for ECDSA signing and verification schemes, key generation and ECDH key agreement schemes for Weierstrass and Edwardian curves. The implementation is protected against Channel Analysis and Fault Injection attacks. The combination of engine and driver provide functionality required by FCS_COP.1/ECC "Cryptographic operation – ECC", FCS_CKM.1/ECC "Cryptographic key generation – ECC", and FCS_CKM.4/ECC "Cryptographic key destruction - ECC".

8.15. KDF (SF_KDF)

The TOE provides a KDF driver for Composite Software usage. The driver uses the AES hardware engine. The TOE is capable of derivation of unique keys originated from the HUK. The combination of AES engine and KDF driver provide functionality required by FCS_CKM.1/KDF "Cryptographic key generation – KDF", and FCS_CKM.4/KDF "Cryptographic key destruction - KDF".

8.16. TOE Summary Specification Rationale

TOE security function Security Functional Requirement	SF_CONF	SF_INT	SF_PHYS	SF_SC	SF_AC	SF_EM	SF_AM	SF_DUI	SF_LC	SF_BFU	SF_RNG	SF_AES	SF_SHA	SF_KDF	SF_ECC
FRU_FLT.2/Env			X				X								
FPT_FLS.1/Env			X				X								
FRU_FLT.2/Log			X				X								
FPT_FLS.1/Log			X				X								
FMT_LIM.2/Test									X						
FMT_LIM.1/Test									X						
FMT_LIM.2/Debug									X						
FMT_LIM.1/Debug									X						
FAU_SAS.1								X	X	X					
FDP_SDI.2	X	X				X									
FDP_SDC.1	X	X				X									
FPT_PHP.3			X			X									
FPT_ITT.1	X		X	X											
FDP_ITT.1	X		X	X											
FDP_IFC.1	X			X	X										
FCS_RNG.1/TRNG											X				
FCS_RNG.1/DRBG											X				
FCS_COP.1/AES												X			
FCS_COP.1/SHA													X		
FCS_COP.1/ECC															X
FCS_CKM.1/AES												X			
FCS_CKM.1/ECC															X
FCS_CKM.1/KDF														X	
FCS_CKM.4/AES												X			
FCS_CKM.4/ECC															X
FCS_CKM.4/KDF														X	
FMT_LIM.2/Loader									X	X					
FMT_LIM.1/Loader									X	X					
FDP_ITC.1	X					X				X					
FPT_RPL.1/PM						X									
FDP_URC.1/PM						X				X					
FDP_IRA.1/Data						X									

TOE security function Security Functional Requirement	SF_CONF	SF_INT	SF_PHYS	SF_SC	SF_AC	SF_EM	SF_AM	SF_DUI	SF_LC	SF_BFU	SF_RNG	SF_AES	SF_SHA	SF_KDF	SF_ECC
FDP_IRA.1/Code						X									
FDP_DAU.2/PM					X	X				X					
FIA_UID.1/PM					X	X				X					