Secure Sub-System in System-on-Chip (3S in SoC) Protection Profile

Version 1.0

27.05.2021

Registered and Certified by

Federal Office for Information Security (BSI)

under the reference ######

Developed by Eurosmart Members:

Brightsight, Deutsche Telekom Security, Giesecke+Devrient, Infineon, Internet of Trust, JTSec, NXP, Qualcomm, Samsung, STMicroelectronics, Synopsys, Thales, Tiempo-Secure, TrustCB, Winbond, Xilinx

EUROSMART
The Voice of the Digital Security Industry

www.eurosmart.com @Eurosmart_EU @Eurosmart

Rue de la Science 14b | B-1040 Brussels | Belgium
Tel +32 2 880 3635 | mail contact@eurosmart.com

This page will be omitted from the final version.
Use it to track the development of this draft.

Version	Date	Changes
1.0	27.05.2021	First Draft

Table of Contents

1 PP Introduction

1.1 PP Reference

Title	Secure Sub-System Platform Protection Profile
Version:	1.0
Date:	26.05.2021
Developer:	Eurosmart
Technical Editor:	Deutsche Telekom Security GmbH
Certification Body:	Bundesamt für Sicherheit in der Informationstechnik (BSI)
Certification ID:	BSI-CC-PP-++tbd++

1.2 TOE Overview

1.2.1 TOE Type

The TOE is a Secure Sub-System (3S) implemented as a functional block of a System on Chip (SoC). The TOE implements a processing unit, security components, I/O ports and memories to provide a range of security functionalities covering a defined set of security objectives. The TOE provides its security features and security services isolated from the remaining SoC components, based on physical and/or logical isolation mechanisms. The TOE may rely on external memories to store content (data, code or both).

A cohering design within the hosting SoC supports the isolation of the TOE and is a prerequisite for the re-use of the 3S from the initial SoC into other SoCs. The re-use is possible if the interfaces between the 3S and the SoC are preserved and the manufacturing process uses the same technology process in the same production sites.

Interface description and security guidance for the Composite Software development are delivered as part of the TOE, as is documentation for the integration of the 3S.

1.2.2 TOE Definition

The TOE comprises hardware (HW), firmware (FW) and software (SW) required to provide security services and security features. Security services provided by the TOE comprise the functionality of the Root of Trust (RoT) including the unique identification of each instance and the generation of random numbers. Cryptographic functions are defined as optional security services. Security features protect the data stored and processed inside the TOE, as well as support the correct operation of the security services to be provided to the SoC and the "Composite Software[1]". In addition, the TOE includes guidance describing the secure integration into an SoC as well as guidance on configuration and usage or administration operations including update of firmware and software. Any Composite Software shall be isolated from the TOE FW/SW, and the various Composite Software instances shall be isolated from each other. Furthermore, the user data of one Composite Software instance shall not be accessible by another Composite Software instance.

The TOE implements all hardware components required to provide the security services and the protection of the TOE and Composite Software assets. This typically comprises processing unit, volatile

[1] Here, the Embedded Software of a Composite Product executed in the 3S is named "Composite Software". The Composite Software may include parts of the operating systems and one or more applications.

memory, non-volatile memory, communication interfaces, security control circuits, control of power, clock and reset as required for the secure operation and a physical random number generator.

The 3S is a physically fixed design either defined as hard macro (e.g., a GDSII file) and/or as programmable logic (PL) macro (a bitstream used to configure a field programmable gate array). In any case "physically fixed design" means that the layout, placement, routing and timing are part of the implemented 3S, and that the HW implementation is predictable in terms of operational ranges such as performance, timing, area, and power. For a PL Macro, the functionality that the PL Macro provides may be configurable; this configurability shall be independent of the PL Macro placement and routing such that the predictability of operational ranges is not affected.

The 3S is implemented in a System on Chip (SoC) as an independent functional block isolated from the rest of the SoC.

The 3S may have dedicated interfaces to interact with other components of the SoC or with the external world from SoC perspective. These interfaces allow the TOE to obtain information from, or to provide services to other SoC components, Composite Software and external world.

Application Note 1. The TOE may have bi-directional interactions with other SoC components through well identified interfaces, without security dependencies on the other SoC components. If a specific implementation introduces dependencies between the TOE and other SoC components that impact the security functionality, such dependencies shall be described in the Security Target together with associated security requirements if/as needed.

Application Note 2. The 3S is considered to be a monolithic IP block in this Protection Profile, its implementation may be distributed across the SoC. Such specific case is not addressed in this Protection Profile and the specificities of a distributed 3S shall be described in the Security Target together with additional necessary security requirements.

The 3S may include FW/SW stored in a Read Only Memory (ROM). This ROM and its ROM code are part of the TOE.

The Firmware (FW) delivered as part of the TOE includes initialization and secure boot of the TOE and may also include related drivers. Software (SW) may provide additional functionality such as APIs for crypto services and/or other support functions.

The 3S may use memory outside the 3S. In this case the memory is defined as external memory. The protection of the data in the external memory and the link to this external memory can either completely rely on the security functionality implemented in the 3S, or the external memory can implement security functionality supporting the protection of data stored in the external memory and supporting the protection of the link between the 3S and the external memory. In the latter case, the external memory and its interface with the 3S are part of the TOE.

An external non-volatile memory may store a protected instance of the executable SW in this memory. This protected instance of the software is named here a TOE software image. Such software image needs to be loaded in the 3S, authenticated, verified and decrypted by the FW prior to be executed as FW extension or SW. In such cases, Composite Software is also stored in the external non-volatile memory as a specific software image not included in the TOE.

Application Note 3. The distinction between FW and SW from security evaluation point of view is specific for each 3S implementation. The Protection Profile considers the following split between FW and SW: The FW cannot be executed before the hardware is initialised and the SW is securely initialised with the support of the FW. Associated details shall be defined in the Security Target. The terms FW/SW are used throughout the document to capture FW and SW code as well as associated configuration and data.

The TOE implements an initial Root of Trust (firmware + data/keys) that provides security services for the initial phase of the TOE. These security services comprise a secure boot functionality and the authentication, decryption, and verification of TOE software loaded from outside the TOE. An extended Root of Trust (chained from the initial Root of Trust) can be provided to support as well import of keys, certificates and/or data provided by service providers and/or by a composite software developer. The Root of Trust security services support confidentiality, integrity control and authentication when importing code and data in the TOE. The initial Root of Trust is implemented as part of the HW and FW and provides a trusted immutable Security Anchor with unique identification and credentials of each instance of the TOE.

Figure 1 describes the typical interfaces of the TOE in the SoC.

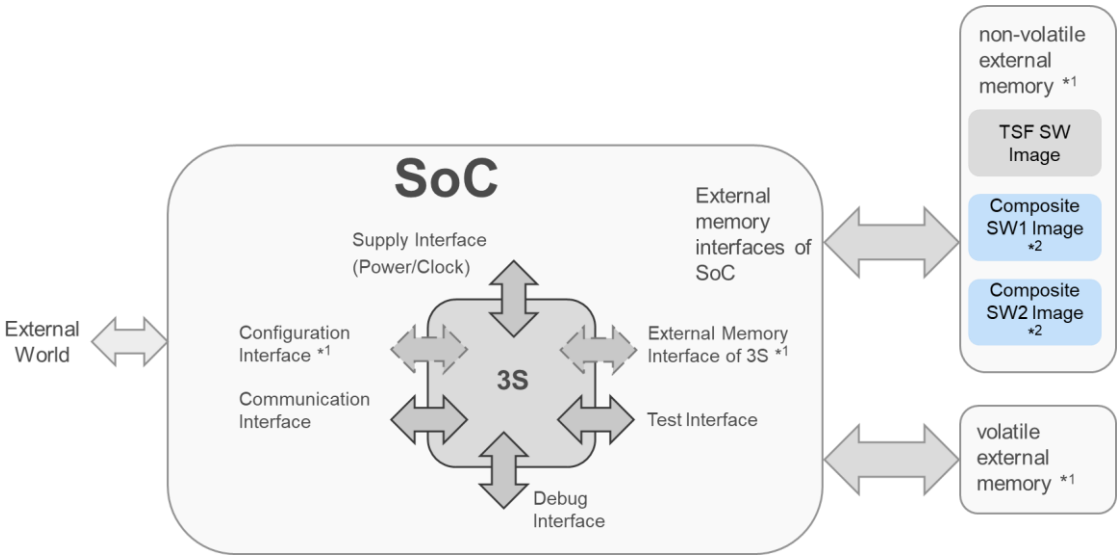


Figure 1: Interfaces of the TOE

*1 3S interfaces marked with dashed lines and the use of the external memories by the 3S are optional, depending on the implementation and configuration of the TOE.

*2 Composite Software Images do not belong to the TOE.

The functionality of the interfaces between the TOE and the SoC shall be clearly described to support the isolation of the TOE. Furthermore, the interfaces shall have a limited complexity with minimum dependency between each other and clearly defined functionality and purpose. This shall support the control of the interfaces and restrict the attack surface of the TOE.

The power supply interface comprises one or more power rails. The 3S may be driven by the clock from the hosting SoC. The 3S can also implement its own clock. The supply interface also comprises the reset signal of the SoC.

The communication interface is intended for the exchange of data between the 3S and the remaining SoC. The implementation of the communication interface shall allow a clear control and separation between the 3S and the SoC.

Application Note 4. The communication interface may include dedicated support for the connection to remote systems or implements an interface that uniformly supports the data exchange with various components of the SoC. The Security Target (ST) author shall supplement the description, based on the specific implementation.

The debug interface and the test interface are limited to development and manufacturing. The circuitry controlling these interfaces shall be completely included in the 3S.

Application Note 5. Additional interfaces (e.g., for configuration purposes) shall be added by the ST author. They may contribute to the life-cycle management of the 3S or allow the enabling or disabling of specific components of the 3S.

Application Note 6. The 3S direct or indirect interfaces with the external world are dependent from the 3S implementation. Details shall be described in the Security Target.

The external memory is optional. In respect to the external memory, the Protection Profile supports different TOE configurations. The base Protection Profile includes the security functionality of the 3S without external memory. The configurations with external memory described in section 7.1 and 7.2 are defined as separate packages, see section 1.3. A Security Target may use none, one or both of these packages, in any combination for volatile and non-volatile memory.

Application Note 7. There are dependencies between different packages (e.g., the Loader Package can be added to the Security Target only if it includes the package external non-volatile memory). For more details, see section 1.3.

Even when confidentiality and integrity of the content stored in the external memory are ensured, a new scenario of threat exists when the content stored in the external memory could be read, stored, and later written back to the external memory. This situation opens the possibility of an unauthorised rollback of the content in the external memory to a previous version. The same effect could be achieved by intercepting communications passing across the interconnection bus between the external memory and the 3S and replaying the replies to previous read commands. Although the content replayed or written to the external memory were valid at a given moment in the past, this attack prevents the TOE from obtaining or updating the latest or “fresh” version of the content in the external memory.

The freshness of content qualifies the property that stored content are always the one resulting in the last change carried out by the 3S on the external memory. An attack consisting of replacing the content in the external memory with a previous version (e.g., cloning at a given time), which would result in writing to the external memory content that preserves its confidentiality, integrity, and authenticity, would violate the “freshness” of the content. Content stored in the external memory shall also be protected in terms of data freshness.

Application Note 8. The author of the Security Target shall list all interfaces of the 3S. The number and functionality of these interfaces depend on the implementation of the 3S. E.g., the configuration interface may not be available or it may only comprise dedicated wires with fixed signals. As another example the debug interface and test interface may be merged into a single interface.

Application Note 9. For a given implementation, the TOE may have dependencies on the hosting SoC and they shall be described in the integration guidance to enable transferability of the results of the evaluation of a 3S in a given SoC when integrated into another SoC.

1.2.3 Usage and Major Security Features of a TOE

The TOE can be used for multiple application areas that require a high level of security, including:

- User authentication and password storage
- Content protection
- Payment
- Subscriber identity module (SIM)
- Storage and management of digital identities
- Secure key storage
- Root of trust
- Storage of sensitive user data (e.g., healthcare records).

The TOE provides a security service to identify each instance of the 3S and to demonstrate the authenticity of HW and FW.

The Protection Profile defines a basic set of security services and security features that shall be provided by the TOE. The security services and security functionality may be extended to support the additional needs of specific configurations.

This Protection Profile supports the following types of memory:

- memory integrated in 3S inside the TOE perimeter named internal memory (IM)
- External memory outside the TOE perimeter named passive external memory (PM)
- External memory inside TOE perimeter named secure external memory (SM)

The details of the configurations with external memories are described in the related sections defining the associated package. The base Protection Profile comprises the configuration with internal memory (IM) only. This configuration of the TOE includes all memory resources required for the operation of the TOE. The FW and SW are stored inside the memories of the TOE. Optionally a FW/SW image can be downloaded and verified in the TOE during a FW/SW update operation.

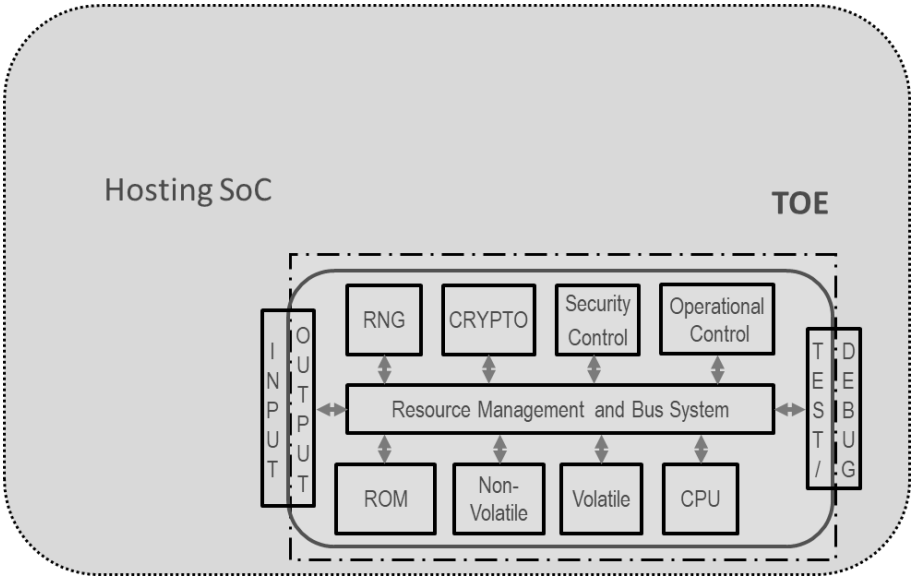


Figure 2: All components are integrated inside the 3S

1.2.4 Required Non-TOE hardware/software/firmware

The hosting SoC provides power supply and associated power management and reset management to operate the 3S. Further on, the SoC may provide clock signals to the 3S, for example. In addition, the hosting SoC supports the interfaces of the TOE to enable communication between the 3S and the hosting SoC. The interfaces of the 3S may be allowed to connect to remote systems via the external interfaces of the SoC. The connection may be used to perform transactions or download updates. The hosting SoC also provides interfaces to external memories or provides additional memory resources on its own. These may be used by the 3S as outlined in section 1.2.2.

Application Note 10. The dependencies on the hosting SoC shall be outlined in the integration guidance.

1.2.5 TOE Life Cycle

The hardware of the 3S needs to be integrated into a hosting SoC. The integration process is applicable if the developers of the 3S and the hosting SoC belong to the same company, or if the 3S developer provides the 3S to an external company.

The integration process needs to ensure the integrity and confidentiality of the hard macro delivered by the 3S developer. All interfaces between the TOE and the SoC shall be used as described in the integration guidance. The hosting SoC may provide power supply and control signals as part of the operational environment for the 3S.

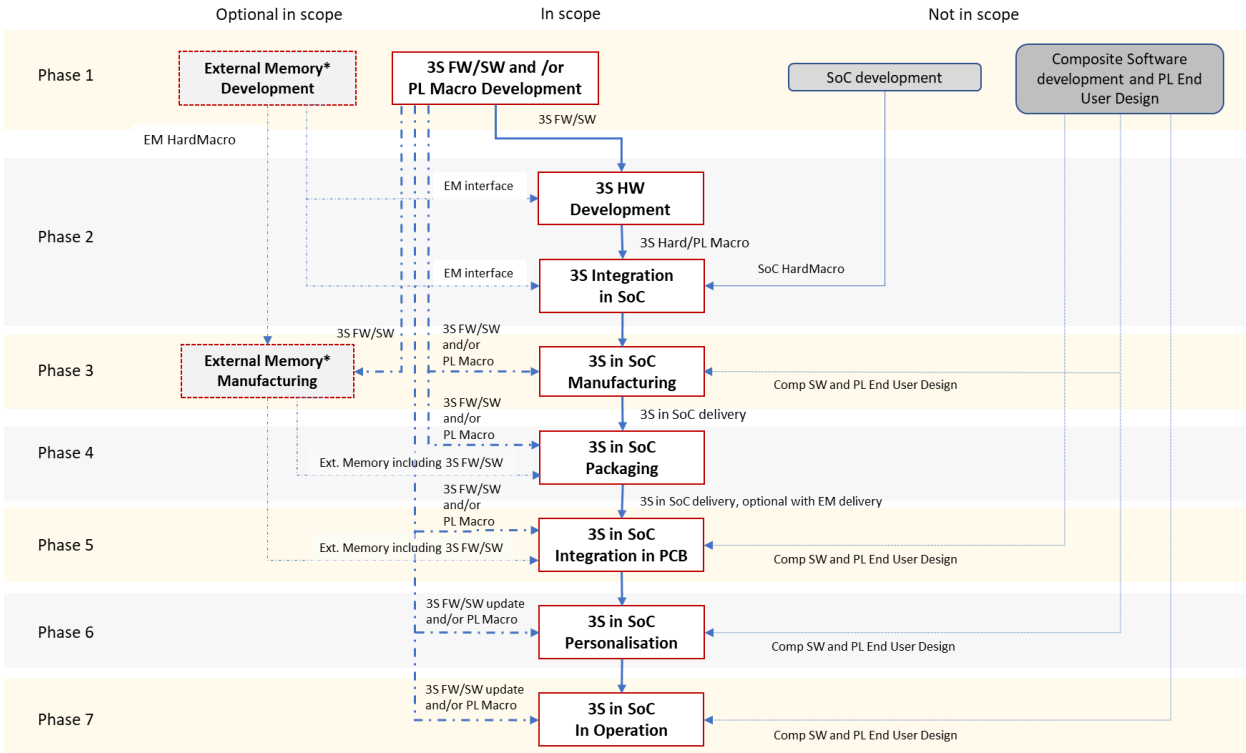
The complex hardware and software development process of System on Chips including a 3S can be split into seven generic phases. The form factor and the integration of the SoC are not standardised. Therefore, the life cycle can depend on the intended usage of the Composite Product. This can comprise the SoC packaging but also the download of the software and Composite Software.

The development of the hard macro is part of Phase 2, as shown in Figure 3. The development of the hosting SoC is also part of Phase 2, because both these developments need to be delivered as one complete product to the wafer fab as part of Phase 3. The development of hardware specific firmware including boot software and drivers are also part of the development in Phase 2, because this software is integrated in the hardware design.

The evaluation of the 3S development environment shall include all life-cycle phases that are required to trim, configure and personalise the 3S. After these steps the self-protection of the 3S shall be enabled and ensure the protection of the TOE. If the trimming, configuration and personalisation is done as part of the wafer test at the end of Phase 3, the delivery can be applied at the end of Phase 3. If the trimming, configuration and personalisation is performed after the IC packaging, Phase 4 needs to be in the scope of the evaluation. The external memory is manufactured in Phase 3. After manufacturing, the firmware and software, as well as composite software, might be loaded to the external memory. In the case firmware, software and/or composite software are stored in the external memory, they should be protected.

The secure external memory can be evaluated as part of the TOE or may have been evaluated separately, with evaluation results re-used during the evaluation of the TOE, based on the composition approach.

The following figure describes the life cycle of the TOE:



* Secure External Memory is in the evaluation scope.

Figure 3: Life Cycle of TOE

Figure 3 describes a typical life cycle with different options of the initial loading and update of FW and SW. All items in dashed lines are optional according to the selected use case. The PL Macro is delivered via the optional path for 3S FW/SW distribution. The development of PL End User Design is independent of the PL Macro development for the 3S and not in the scope of the evaluation. In most cases, the delivery type of the SoC including the 3S is performed at the end of Phase 3 or Phase 4. The SoC development and the development of the Composite Application (Comp APP) are out of evaluation scope.

1.2.5.1 Phase 1: 3S Firmware and Software Development

The TOE SW can be stored either in the 3S or in external non-volatile memory. If the SoC comprises programmable logic also the development of the 3S PL macro is performed in this phase.

Application Note 11. The split of the software development between Phase 1 and Phase 2 depends on the processes defined by the developer of the 3S hardware and software. The details required to define the evaluation scope shall be included in the product specific Security Target.

Phase 1 also includes the design and development of the Composite Software for the 3S. Depending on the configuration, the Composite Software is stored on the 3S or is stored in the external non-volatile memory. If the Composite Software is remotely loaded using a secure loader, this loader shall be in the scope of the evaluation. Based on the use of a secure loader, life-cycle phase or the site where the download is applied are not security relevant.

1.2.5.2 Phase 2: 3S hardware development and integration into SoC

Comprises the development of the 3S hard macro and associated firmware. Phase 2 also comprises the development of the SoC hardware with the interfaces to the 3S. The development of the SoC is not in the scope of the evaluation. The scope of the evaluation for Phase 2 is determined by the transfer of the 3S hard macro to the developer of the hosting SoC.

The deliverables of the 3S development comprise a hard macro and/or a Programmable Logic macro, associated guidance for the integration of the 3S as well as preparation of FW/SW code that is integrated in the ROM of the 3S. The protection of the 3S design has to be ensured by the development environment. The integration of 3S hard macro on the SoC is performed in this life-cycle step. In addition, the 3S can run on a SoC simulation.

The integration of the 3S on the SoC needs to be completed before the complete SoC is delivered to the mask shop or wafer fab that belongs to life-cycle Phase 3. The delivery needs to include all components that are required for production of the SoC including the 3S. This comprises the hardware design of the SoC including the 3S, the FW and the SW. Components of the Security Anchor, as well as credentials for production/preparation required for production, also need to be part of the delivery. The 3S design is protected by limiting the 3S design block to the information required for the integration and by protecting the integration environment of the SoC with the 3S. The transfer of the SoC including the 3S to the production shall protect the confidentiality and integrity of the complete design.

Application Note 12. The integrator that integrates the 3S in the SoC during Phase 2 of the life-cycle is a user of the TOE and as such the integration guidance is a TOE component and shall be assessed during AGD.

1.2.5.3 Phase 3: 3S in SoC Manufacturing

The manufacturing of external memory can be included as option.

The manufacturing comprises the production and the functional testing of the SoC, including the 3S. The tests of the 3S can be mainly independent of the SoC or they may be integral part of the test applied for the SoC. The testing in this phase can also include the initialisation and provisioning/completion of the 3S including the Security Anchor, but this is not mandatory of the TOE.

The initialisation includes the trimming, configuration and provisioning of a unique ID for each functional device.

The testing in this phase can also include the initialisation and personalisation of the TOE.

The scope of the evaluation shall include the complete trimming, initialisation and pre-personalisation of the 3S. The scope of the evaluation can be limited to Phase 3, if these steps are all performed in Phase 3 and the self-protection of the 3S is active at the end of Phase 3.

At the end of Phase 3 also parts of the FW/SW for the 3S can be loaded into internal memories of the 3S. For secure external memory, FW/SW can be stored in the secure external memory at the end of Phase 3.

The exchange of software and scripts between the 3S developer and the test centre required for the testing, initialisation, pre-personalisation and provisioning needs to be described and considered during the evaluation.

The SoC including the 3S can be delivered to the customer at the end of this life-cycle phase. The 3S integrated in the SoC, as well as FW and SW can be delivered together, but this is not mandatory because the external memory may not be integrated in this life-cycle phase.

Application Note 13. The SoC including the 3S can only be considered as delivery item at the end of Phase 3, if the trimming, initialisation and provisioning/completion is completed and the self-protection of the 3S is completely enabled. The evaluation shall include all manufacturing steps, where the trimming, initialisation and provisioning/completion is not finished or the loading of SW or Composite Software which require protection by the environment.

1.2.5.4 Phase 4: 3S in SoC Packaging

The packaging comprises the assembly of the SoC in a package. This may include the stacking of the SoC with memory in the same package. The packaged devices are subsequently tested. These tests also can comprise additional trimming, initialisation and provisioning/completion of the 3S, if this is not completed in Phase 3. In addition, loading of SW or Composite Software can be performed in this life-cycle phase if the trimming, initialisation and provisioning/completion are completed and the required non-volatile memory is already available.

At the end of this life-cycle phase the SoC including the 3S is packaged. This package is ready for the integration on a PCB.

The packaged SoC can be considered as delivery item in the scope of the evaluation, if the self-protection is enabled at the end of Phase 4 and the additional loading of SW or Composite Software on the 3S or in the memory does not require a secure environment.

Application Note 14. The SoC including the 3S can only be considered as delivery item at the end of Phase 4, if the trimming, initialisation and provisioning/completion is completed and the self-protection of the 3S is completely enabled. The evaluation shall include all manufacturing steps, where the trimming, initialisation and provisioning/completion is not finished or the loading of SW or Composite Software requires protection by the environment.

1.2.5.5 Phase 5: 3S in SoC Integration in PCB

The SoC integration in PCB comprises further integration step, as soldering in the PCB. If the self-protection of the 3S is already enabled in preceding life-cycle phases, this phase does not need to be part of the evaluation.

The non-volatile memory may be integrated in this phase, so the SW stored in the external non-volatile memory might initially be downloaded in this life-cycle phase. It depends on the security mechanisms

implemented in the loader of the 3S and security policy of the software, if the loading of the SW requires a trusted environment.

In most cases, this life-cycle phase is performed by various integrators, therefore it is not included in the scope of this protection profile. If required, related guidance needs to be included in guidance documentation of the TOE.

1.2.5.6 Phase 6: 3S in SoC Personalisation

Phase 6 is the personalisation phase that may also include customer specific configuration of the 3S. The 3S developer may leave configuration tasks to the personaliser. Such tasks are considered to be part of the preparative guidance for the 3S. In this personalisation phase an authorised user can perform an optional update of the 3S FW or SW. The user may be the administrator of this life-cycle phase.

1.2.5.7 Phase 7: 3S in SoC in Operation

Phase 7 is the operational phase, where the administrator operates the 3S in SoC and the end-user uses the device including the 3S in SoC.

Application Note 15. The developer of the 3S can determine which life-cycle phases are in the scope of the evaluation. This is limited, however, depending on the implementation of the Test Mode and the implementation of the trimming, initialisation and provisioning/completion. The life-cycle phases of the 3S need to be in the scope of the evaluation as long as Test Mode is enabled and may be misused (e.g., for characterisation purposes) and/or the trimming, initialisation and provisioning/completion includes assets (e.g., a unique ID or key splits or private/public keys that need to be protected by the environment).

1.3 Functional Packages

This Protection Profile includes several optional packages to extend the security functionality of the base Protection Profile including the use of external memory. For details, see Chapter 7.

Each package defines a specific security problem, a set of security objectives and the corresponding Security Functional Requirements (SFRs).

The configurations with external memory are defined as packages. If the 3S is connected to an external memory, the package associated with the type of external memory shall be added in the Security Target (ST).

The packages not related to the external memory are applicable to all memory configurations. The functionality and complexity of the SoC that hosts the 3S is independent from the functionality of the 3S.

The following figure illustrates the packages defined in this PP:

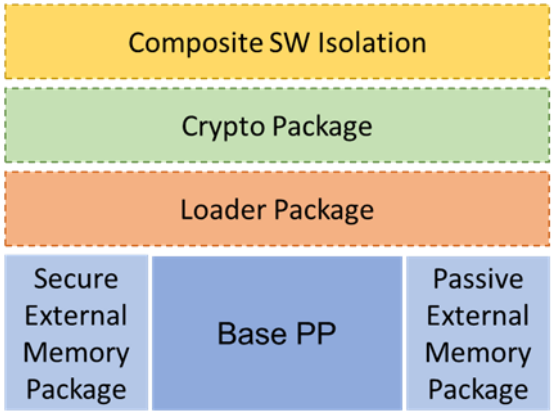


Figure 4: Package structure of this Protection Profile

Package Name	Package Purpose	Reference	Relationship
Base PP		Section 3 to 6	Mandatory
Passive External Memory Package	The 3S is connected to a passive external memory. Neither the passive external memory nor the connection between the passive external memory and the 3S provide protection for software and data. The 3S shall protect software and data before it is transferred from or to the passive external memory.	Section 7.1	Optional
Secure External Memory Package	The 3S is connected to a secure external memory. The secure external memory protects stored code and data. In addition, the 3S and the secure external memory implement security mechanisms to protect the exchange of code and data.	Section 7.2	Optional
Loader Package	Loading of 3S SW or Composite Software from external memory. The package defines rigorous security functionality to restrict the loading of authenticated images with integrity protection prior to the execution by the TOE.	Section 7.3	Optional Dependency on external NVM package(s)
Crypto Package	The package provides a framework for the integration of various cryptographic algorithms supported by the TOE.	Section 7.4	Optional
Composite Software Isolation Package	The isolation features provided by the hardware and the FW/SW of the 3S implement self-protection and separation between the FW/SW belonging to the 3S and the Composite Software instances.	Section 7.5	Optional

Table 1: Overview of the functional packages

2 Conformance Claims

2.1 CC Conformance Claim

This Protection Profile claims to be conformant to the Common Criteria (CC), Version 3.1, Revision 5.

Conformance of this PP with respect to CC Part 2 (security functional components) is CC Part 2 extended, see [2].

Conformance of this PP with respect to CC Part 3 (security assurance components) is CC Part 3 conformant, see [3].

2.2 PP Claim

This PP claims strict conformance to the Protection Profile Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, 13.01.2014, registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014, see [8].

The 3S is intended as Security IC integrated in SoC. The 3S provides at least the same security services to Composite Software as a Security IC. Functional packages can be added to extend the security services and support different memory configurations.

2.3 Package Claim

The minimum assurance level for this Protection Profile is EAL4 augmented with ATE_DPT.2[2], AVA_VAN.5 and ALC_DVS.2.

2.4 Conformance Rationale

The TOE type is a 3S comprising a processing unit, security components, I/O ports and memories. This TOE type is intended as platform providing security services. This applies for BSI-CC-PP-0084-2014 as well as for this 3S in SoC PP. The security IC defined in BSI-CC-PP-0084-2014 is a dedicated device while the 3S defined in this PP is a physical and/or logical isolated component that is integrated into a SoC.

The conformance rationale requires a detailed analysis of the security problem definition, the security objectives, the security requirements and the threats defined in BSI-CC-PP-0084-2014 and in the PP in hand. These details are moved to section 9.1 of this Protection Profile.

2.5 Conformance Statement

The Protection Profile requires strict conformance of the Security Target or Protection Profile claiming conformance to this Protection Profile.

[2] ATE_DPT.2 – This PP claims the same assurance packages as claimed in PP0084.

3 Security Problem Definition

3.1 Description of Assets

The assets of the TOE are:

- User data of the TOE and the user data of the Composite Software[3].
- 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 PL Macro, if the 3S is at least partly implemented with programmable logic.

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

SC1: integrity and authenticity of user data,

SC2: confidentiality of user data of the TOE and the Composite TOE being stored in the TOE's protected memory areas,

SC3: correct operation of the security services including the root of trust provided by the TOE for the Composite Software.

The Composite Software is user data and shall be protected while being executed/processed and while being stored in the TOE's protected memories.

The TOE may not distinguish between user data which is publicly known or kept confidential. Therefore, the TOE supports the protection of the user data in integrity, authenticity and confidentiality if stored in protected memory areas, unless the Composite Software chooses to disclose or modify it.

The integrity and authenticity of the software including Composite Software means that it is correctly being executed. This includes especially the correct operation of the TOE's security services including the root of trust. Parts of the FW, SW and Composite Software that do not contain secret data or security critical source code, may not require protection from being disclosed. Other parts of the FW, SW and Composite Software may need to be kept confidential, because specific implementation details may assist an attacker.

The TOE Manufacturer shall apply protection to support the security of the TOE. This applies to the TOE and to all information and material exchanged with the developer of the Composite Software. This covers the Composite Software itself or any authentication data required to enable the installation of software in the TOE, including in phases after TOE Delivery.

The TSF processes user data objects (code and/or data) as well as TSF data objects. User data objects are imported, used in cryptographic operation, temporarily stored, exported and may be destroyed after use. They may contain cryptographic keys with or without security attributes, certificates and authentication data of a device/user. Cryptographic keys are objects of the key management.

Application Note 16. The limitation of the protection provided by the different memories of the 3S for the Composite Software need to be detailed in the Security Target and the User Guidance associated with the TOE.

[3] The Composite Software as well as the User Data of the Composite Software are both considered as part of the User Data of the TOE. The TOE, however, may allow different protection mechanisms for code and data. Therefore, they are mentioned separately in the assets.

Application Note 17. Wide-ranging protection mechanisms may be applied for TSF data as well as user data. This may comprise splitting or masking of confidential information. In such case the protection of the confidentiality is considered to be ascertained as long as any revealed part of the data is not sufficient to reveal the secret under high attack potential.

Application Note 18. As long as the user data of the TOE or of the Composite Software is unique it can be protected more effectively compared to the FW, SW and Composite Software that is the same for all instances of the TOE. If specific security mechanisms providing additional protection of Firmware, Software or Composite Software (or at least to parts of these software components) are implemented, this shall be detailed by the ST author.

3.2 Threats

The threats described in this section are applicable to the base Protection Profile. For threats related to functional extensions see Chapter 7.

The following figure describes the attacks that are applicable to the TOE. The interactions related to the attacks are marked with red arrows.

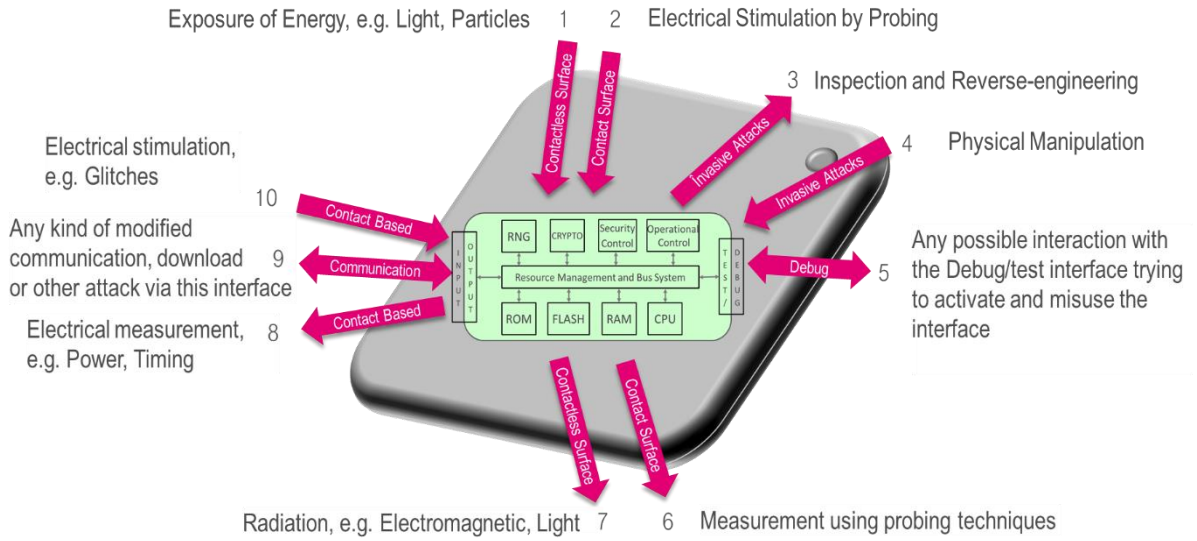


Figure 5: Attacks against the TOE

The Grey box represents the SoC and the green box represents the 3S. The 3S comprises various interfaces (see Figure 1), the dedicated interfaces are named in the threat description. Attacks may be applied on the internal interface between the 3S and the SoC or attacks may be applied from outside the SoC if an interface of the SoC is directly connected to the 3S. This depends on the implementation of the 3S. E.g., exposure to light is directly applicable to the 3S because it is part of the SoC substrate, while direct probing is possible only if the 3S uses all metal layers of the design. For the communication interface it depends whether remote connections are directly routed to the 3S or whether parts of the protocol stack are included in an application running on the SoC.

The surface of the 3S does not provide an interface from a functional point of view, but it is considered to be an interface for an attacker.

The TOE shall avert the threat “Inherent Information Leakage (T.Leak-Inherent)”, as follows:

T.Leak-Inherent Inherent Information Leakage

	An attacker may exploit 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.

Leakage may occur through emanations, variations in power consumption, response times, clock frequency, or similar variations in the behaviour, based on the data processed by the TOE. This leakage is related to measurement of operating parameters, which may be derived either from measurements of internal and/or external supply signals and/or measurement of emanations and/or IO signal. These operating parameters can then be matched to the specific operations inside the TOE. Examples of such attacks are Differential Power Analysis and Timing Attacks (8 in Figure 5), or analysis of emanation (7 in Figure 5).

The leakage may also be generated by the hosting SoC. It may not be possible to split between the power analysis of the TOE and of the SoC. This may make an attack more difficult but does not prevent attacks. Inherent emanation leakage may be identifiable also outside the TOE boundaries on the surface of the SoC and does not require direct contact with TOE internal signals.

The TOE shall avert the threat “Physical Probing (T.Phys-Probing)”, as follows:

T.Phys-Probing	Physical Probing
	An attacker may perform physical probing of the TOE. The probing is performed (i) to disclose user data or TSF data while stored in protected memory areas, (ii) to disclose/reconstruct user data or TSF data while processed or (iii) 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.

Physical probing requires direct interaction with the hardware of the TOE inside the TOE boundary or at the border of the TOE. Physical probing done at the SoC level may also be used, however, to gain knowledge of the TOE.

Techniques and tools commonly employed in failure analysis and reverse engineering may be used for such attacks (2 and 6 in Figure 5). Before hardware security mechanisms and layout characteristics can be attacked, they need to be identified by reverse engineering. The analysis of software behaviour or processing of user data or TSF data may also be a prerequisite for the attack.

The TOE shall avert the threat “Malfunction due to Environmental Stress (T.Malfunction)” as specified below.

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 (i) modify security services of the TOE or (ii) modify Composite Software including composite user data while being processed by security services of the platform, or (iii) deactivate or affect the TSF to enable disclosure or manipulation of user data. An attacker may also cause malfunction by (iv) modifying data or messages, or by (v) misuse of architectural and micro architectural weaknesses via control and communication interfaces.

The environmental stress can either directly be applied to the TOE or introduced via the interfaces of the SoC that integrates the 3S. The attacker may apply the environmental stress to the SoC without knowledge of details regarding the location and interaction between the TOE and the SoC hosting the

3S. Beside the environmental stress also logical attacks can cause malfunctions and impact the security features and security services.

The modification of security services of the TOE may affect the quality of random numbers provided by the random number generator, the malfunction of cryptographic coprocessors or the manipulation of TSF data or user data stored in the volatile memory. An attacker needs information about the functional operation. Based on this information the attacker can introduce a temporary failure by exposing energy to the 3S (1 in Figure 5) or (10 in Figure 5). This may be achieved by operating the TOE outside the normal operating conditions. The same attack techniques applied at SoC interfaces level could also provoke malfunction of the TOE.

Modification of security services, circumvention of access control or forced leakage may also be achieved by exploiting physical, architectural or micro architectural weaknesses at the interfaces of the 3S, or disturbing or modifying the communication (9 in Figure 5) between the SoC and the 3S, or exposure of energy (1 in Figure 5) or glitches on the interfaces (10 in Figure 5) causing errors that lead to an exploitation of these weaknesses.

The TOE shall avert the threat “Physical Manipulation (T.Phys-Manipulation)” as specified below.

T.Phys-Manipulation	Physical Manipulation An attacker may physically modify the TOE or the SoC, to (i) modify user data of the Composite Product, (ii) modify the Composite Software, (iii) modify or deactivate security services of the TOE, or (iv) modify TSF of the TOE to enable attacks disclosing or manipulating TSF data, user data or the Composite Software.

The modification may be achieved through techniques commonly employed in failure analysis and reverse engineering efforts (numbers 3 and 4 in Figure 5). The modification may result in the deactivation of a security features. To apply this attack, the hardware security mechanisms and layout characteristics need to be identified. Determination of software design including treatment of user data of the Composite Product may also be a prerequisite. Changes of circuitry or data can be permanent or temporary. Some physical manipulations done at the SoC level could be used to gain knowledge of the TOE.

The TOE shall avert the threat “Forced Information Leakage (T.Leak-Forced)“ as specified below.

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.

This threat pertains to attacks where environmental stress or physical manipulation is applied to the TOE or the hosting SoC to cause leakage from signals which do not compromise user data or TSF data during normal operation. This threat pertains to attacks where methods described in “Malfunction due to Environmental Stress” (see T.Malfunction) and/or “Physical Manipulation” (see T.Phys-Manipulation) are used to cause leakage from signals (Numbers 5, 6, 7, 8 or 9 in Figure 5) that normally do not contain significant information about secrets.

The threat also covers any influence of the SoC (e.g., by modification of the power management causing environmental stress without glitching or physical manipulation). Such threats may also force leakage of significant information about assets processed by the TOE. The same attack techniques applied at SoC interfaces level could also result in disclosure of sensitive data.

The TOE shall avert the threat “Abuse of Functionality (T.Abuse-Func)” as specified below.

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 (i) disclose or manipulate TSF data or user data, (ii) manipulate (explore, bypass, deactivate or change) security services of the TOE or (iii) manipulate (explore, bypass, deactivate or change) functions of the TOE FW/SW and of the Composite Software, or (iv) enable an attack disclosing or manipulating user data or the Composite Software.

This threat comprises the misuse of test and debug functionality provided by the TOE (5 in Figure 5). Further on an attacker may misuse or manipulate functions intended for the configuration and life-cycle control of the TOE. This can comprise one or more interfaces either between the TOE and the SoC or interfaces providing external access to the TOE. Conducting attacks through SoC debug or tests interfaces could also have an impact on the TOE protection.

The TOE shall avert the threat “Deficiency of Random Numbers (T.RND)” as specified below.

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.

This threat addresses the analysis of random numbers produced by the TOE security services under the various conditions under the control of an attacker. Unpredictability is the main property of random numbers, so this may be a problem if they are used to generate cryptographic keys or blinding parameters, for example. The entropy provided by the random numbers shall be appropriate for the strength of the cryptographic algorithm, the key, the cryptographic variable (e.g., masking) they are used for. Here the attacker is expected to take advantage of statistical properties of the random numbers generated by the TOE. Malfunctions or premature ageing are also considered which may assist in getting information about random numbers. The attack applies to random numbers used by the TOE or provided by the TOE as security services.

The TOE shall avert the threat “Insecure State of the TOE (T.Insecure-State)” as specified below. An insecure boot process can occur during attacks, such as error manipulation of the TOE or hosting SoC manipulation that impacts the boot process. The attack may lead to a wrong initialisation of security services or security features, or the acceptance, import and execution of hostile 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.Phys-Manipulation during start-up, which may force malicious code execution or TSF data manipulation. In this way, an attacker may (i) force invalid settings of the TOE hardware (e.g., life-cycle state, trimming, etc.), (ii) load and execute unauthenticated firmware and/or software, (iii) masquerade the unique identity, or (iv) archive an inconsistent initialisation of the Root of Trust in order to compromise secrets or enable other threats.

This threat attacks the secure operation of the TOE and the TOE specific initialisation and configuration during start-up. The initialisation of Root of Trust during the boot process also may be violated by an attacker (see T.Malfunction and T.Phys-Manipulation for applicable attack technics).

3.3 Organisational Security Policies

This section describes the policies applied in this Protection Profile.

The following organisational security policies need to be applied.

Either the 3S Developer or the 3S Integrator shall apply the policy “Generation of device’s individual identifier (P.Gen-Unique-ID)” as specified below.

P.Gen-Unique-ID: Identification of each TOE instance

An accurate identification shall be established for the TOE. The policy requires that each instantiation of the TOE stores its own unique identification.

A unique identification shall be stored on each instance of the TOE. The testing, trimming and configuration of the TOE after production shall include the download of the unique identification. These processes are in the evaluation scope of the life cycle and performed before the TOE is delivered. The unique identification also considers that the TOE may be delivered to different 3S integrators performing their own configuration and trimming of the TOE.

3.4 Assumptions

The following section describes the assumptions applied in this Protection Profile.

The stacking of additional components in a common packaging may provide additional protection and shielding to the 3S included in the SoC (e.g., countermeasures, such as a metal mesh sensor). If the final assembly and packaging is done after delivery (e.g., by an OEM) and/or after pre-personalisation, the following optional assumption shall be added:

A.Packaging-Requirement: Requirements on packaging

It is assumed that the packaging manufacturer follows the packaging design specifications provided by the 3S developer so the final packaging contributes to the protection and shielding of the 3S in SoC.

Application Note 19. If the packaging shall be included in the evaluation scope and the assessment, this optional assumption needs to be added and the final packaging shall be described in the life cycle section of the Security Target. Additional components (such as dedicated DDR memory) may be added to the SoC (e.g., using a Package-on-Package or other forms of manufacturing integration mechanisms). In most cases, this manufacturing integration step is performed after the pre-personalisation and delivery of the SoC including the TOE. The evaluation of the platform requires an assessment if the operational user guidance sufficiently describes the security measures for the operational environment. Based on this assumption and the related security objective for the environment, the evaluation of the packaging specification is considered to be part of this assessment.

Application Note 20. In this context, the final packaging shall be described in the life cycle section of the Security Target. Additionally, external memory may be added after delivery of the TOE (e.g., using a Package-on-Package (POP) or other forms of manufacturing integration mechanisms), after the TOE has been pre-

personalised, up to delivery of the device including the TOE. In such cases, external memory assembly and integration processes shall also be described in the life-cycle section of the Security Target. All passive external memory, however, is not considered to be part of the TOE.

Appropriate "Protection during Packaging, Finishing and Personalisation (A.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.

A.Process-Sec-IC: Protection during Packaging, Finishing and Personalisation

It is assumed that security procedures are in place after delivery of the TOE (3S included in the SoC) up to delivery of the device to the end-user to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (including the prevention of any possible copy, modification, retention, theft or unauthorised use).

The protection of the TOE is required until the delivery of the product (including the TOE) to the end-user. The assembly and integration processes are part of the evaluated life-cycle scope until the initialisation and pre-personalisation is completed. The TOE needs to be controlled and protected, however, until it is delivered to the end-user.

Application Note 21. The Security Target shall describe the initialisation and pre-personalisation steps covered in the scope of the evaluation.

The Composite Software shall ensure the appropriate "Treatment of user data of the Composite Product (A.Resp-Appl)" as specified below.

A.Resp-Appl: Treatment of user data of the Composite Product

It is assumed that user data of the Composite Product is owned by the Composite Software and treated as required for the specific application context if processed by the Composite Software. Therefore, the Composite Software shall fulfil the guidance of the 3S when security relevant code of the Composite Software is executed and/or security relevant user data of the Composite Product is processed by the Composite Software (especially cryptographic keys).

The application context specifies how the user data of the Composite Product shall be handled and protected. The evaluation of the 3S HW, FW and SW according to this Protection Profile is conducted on generalised application context. The concrete requirements for the Composite Software shall be defined in the Protection Profile [respective Security Target] of the Composite Product. The 3S cannot prevent any compromising or modification of user data of the Composite Product by malicious Composite Software.

4 Security Objectives

This chapter describes the security objectives.

4.1 Security Objectives for the TOE

The user has the following high-level security goals related to the assets:

SG.1 maintain the integrity of user data (when being executed/processed and when being stored in the TOE's memories)

SG.2 maintain the confidentiality of user data (when being processed and when being stored in the TOE's protected memories).

SG.3 maintain the correct operation of the security services provided by the TOE for the Composite Software.

SG.4 maintain the authenticity of the boot sequence and the setup of the root of trust.

SG.5 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, 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. Parts of the Composite Software which do not contain secret data or security critical source code, may not require protection from being disclosed. Other parts of the Composite Software may need to be kept confidential because specific implementation details can assist an attacker.

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.

The TOE shall provide "Protection against Inherent Information Leakage (O.Leak-Inherent)" as specified below.

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 (i) 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 (ii) by measurement and analysis of the time between events found by measuring signals (e.g., on the power, clock, or I/O lines).

This objective pertains to measurements with subsequent complex signal processing whereas O.Phys-Probing is about direct measurements on elements on the chip surface. Details correspond to an analysis of attack scenarios, which are not given here.

The TOE shall provide “Protection against Physical Probing (O.Phys-Probing)” as specified below.

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 (i) 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 (ii) 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.

The TOE shall be designed and fabricated so that it requires a high combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information or other information which could be used to compromise security through such a physical attack.

The TOE shall provide “Protection against Malfunctions (O.Malfunction)” as specified below.

O.Malfunction	Protection against Malfunctions The TOE shall ensure its correct operation. The TOE shall indicate or prevent its operation outside the normal operating conditions where reliability and secure operation has not been proven or tested to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields. Further on, the TOE detects abnormal interface behaviour and/or protocol parameters or protocol sequences that do not meet the specified behaviour.

Remark: A malfunction of the TOE may also be caused using a direct interaction with elements on the chip surface. This is considered as being a manipulation (see O.Phys-Manipulation) provided that detailed knowledge about the TOE´s internal construction is required and the attack is performed in a controlled manner.

The TOE shall provide “Protection against Physical Manipulation (O.Phys-Manipulation)” as specified below.

O.Phys-Manipulation	Protection against Physical Manipulation The TOE shall provide protection against manipulation of the TOE hardware, software and data including FW, SW, TSF data, the Composite Software and the user data of the Composite Product. This comprises protection against (i) reverse-engineering (understanding the design and its properties and functions), (ii) manipulation of the hardware, security services and any sensitive data, as well as (iii) undetected manipulation of TOE memory content.

The TOE shall be designed and fabricated so that it requires a high combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information or other information which could be used to compromise security through such a physical attack.

The TOE shall provide “Protection against Forced Information Leakage (O.Leak-Forced)“ as specified below:

O.Leak-Forced	Protection against Forced Information Leakage The 3S shall be protected against disclosure of confidential user data or TSF data processed or stored in the 3S (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker (i) by forcing a malfunction (see “Protection against Malfunction due to Environmental Stress (O.Malfunction)” and/or (ii) by a physical manipulation (see “Protection against Physical Manipulation (O.Phys-Manipulation)”.

If the protection against forced information leakage is not effective, signals that normally do not contain significant information about secrets could become an information channel for a leakage attack.

The TOE shall provide “Protection against Abuse of Functionality (O.Abuse-Func)” as specified below.

O.Abuse-Func	Protection against Abuse of Functionality The TOE shall prevent functions of the TOE that may not be used after TOE Delivery from being abused and forced to (i) disclose critical TSF data or user data of the Composite Product, (ii) manipulate critical TSF data or user data of the Composite Product, (iii) manipulate Composite Software, or (iv) bypass, deactivate, change or explore security features or security services of the TOE. This also comprises the protection of Test features and/or Debug features provided by the HW, FW and SW of the 3S, which support the development and production of the TOE.

The TOE shall provide “Random Numbers (O.RND)” as specified below.

O.RND	Random Numbers The TOE shall detect and/or prevent manipulation or influence of the entropy source to ensure cryptographic quality of random number generation.

The TOE will ensure that no information about the produced random numbers is available to an attacker, because they might be used to generate cryptographic keys, for example. Further on, the TOE protects the random number generator against manipulation and influence that decrease the entropy of the RNG.

The TOE shall provide “Secure start-up and re-start (O.Secure-State)” as specified below.

O.Secure-State	The TOE shall be started through a secure initialisation process that ensures (i) integrity and authenticity of code executed during start-up, (ii) integrity and authenticity of the hardware settings and the initialisation during start-up including the secure start-up of the Root of Trust functionality.

The TOE shall provide “TOE Identification (O.Identification)“ as specified below:

O.Identification	TOE Identification The TOE shall provide means to store a unique identifier that allows the unique identification of the TOE. Further on, the TOE shall be able to store further initialisation data and pre-personalisation data in non-volatile memory. The unique identifier, the initialization data and the pre-personalisation data are protected against modification.

4.2 Security Objectives for the Environment

The Security Objectives for the Environment are split according to the different life-cycle phases.

4.2.1 Security Objectives for the Composite SW and PL Macro Development (Phase 1)

The development of the Composite Software is outside the development and manufacturing of the TOE. The Composite Software defines the operational use of the TOE. This section describes the security objective for the Composite Software.

Note, to ensure that the TOE is used in a secure manner, the Composite Software shall be designed so that the requirements from the following documents are met: (i) hardware data sheet for the TOE, (ii) data sheet of the Firmware (FW), Software (SW) and the PL Macro of the TOE, and (iii) TOE application notes and other guidance documents that are included in the evaluation of the TOE.

Note that findings of the TOE evaluation need to be addressed in the guidance for the development of Composite Software.

The Composite Software shall provide “Treatment of user data of the Composite Product (OE.Resp-Appl)”, as specified below.

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.

E.g., the Composite Software will not disclose security relevant user data of the Composite Product to unauthorised users or processes when communicating with the remaining SoC or SoC external entities.

4.2.2 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).

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.

Phases after the initialisation can use the individual identifier for tracking and further provisioning. Depending on the application context, the tracking may not be possible in the operational phase of the 3S.

4.2.3 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.

OE.Process-Sec-IC	Protection during Composite Product manufacturing Security procedures shall be applied 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 unauthorised use).

This means that phases after TOE Delivery up to the end of Phase 5 shall protect the TOE appropriately.

4.2.4 Security Objectives for the Operational Environment of the Packaging

Application Note 22. In case the packaging supports the protection of the 3S, this optional security objective for the environment shall be added together with the associated assumption A.Packaging-Requirement

Appropriate "Packaging of the TOE (OE.Packaging-Requirement)" shall be ensured to guarantee the supportive protection of the 3S included in the SoC.

OE.Packaging-Requirement	Packaging of the TOE The stacking, assembly, and packaging of the 3S included in the SoC shall be performed according to the design specification provided by the 3S developer to ensure the additional protection of the 3S by the packaging.

Application Note 23. The design specification of the packaging shall be provided by the 3S developer as part of the guidance delivered together with the TOE.

4.3 Security Objectives Rationale

	O.Leak-Inherent	O.Phys-Probing	O.Malfunction	O.Phys-Manipulation	O.Leak-Forced	O.Abuse-Func	O.RND	O.Secure-State	O.Identification	OE.Resp-Appl	OE.Secure-Initialisation	OE.Process-Sec-IC	OE.Packaging-Requirement
T.Leak-Inherent	X												
T.Phys-Probing		X											
T.Malfunction			X										
T.Phys-Mani-pulation				X									

	O.Leak-Inherent	O.Phys-Probing	O.Malfunction	O.Phys-Manipulation	O.Leak-Forced	O.Abuse-Func	O.RND	O.Secure-State	O.Identification	OE.Resp-Appl	OE.Secure-Initialisation	OE.Process-Sec-IC	OE.Packaging-Requirement
T.Leak-Forced			X	X	X								
T.Abuse-Func						X							
T.RND							X						
T.Insecure-State								X					
P.Gen-Unique-ID:									X		X		
A.Resp-Appl										X			
A.Process-Sec-IC												X	
A.Packaging-Requirement													X

Table 2: Security Objectives versus Assumptions, Threats and Policies

T.Leak-Inherent is countered by O.Leak-Inherent, because the objective requires the protection of confidential TSF data and user data against leakage while being processed and/or stored by the TOE.

T.Phys-Probing is countered by O.Phys-Probing, because the objective requires protection against disclosure and reconstruction of user data or TSF data while stored in protected memory areas and processed by the TOE. In addition, protection is required for disclosure of other critical information about the operation of the TOE.

T.Malfunction is countered by O.Malfunction, because the objective requires indication of operation outside reliable and secure operating conditions or prevent the operation outside the normal operating conditions. Further on, the objective requires the detection of abnormal interface behaviour and protocol parameters or protocol sequences that do not meet the specified behaviour.

T.Phys-Manipulation is countered by O.Phys-Manipulation, because the objective requires protection against manipulation of the TOE comprising TOE hardware, software including FW, SW, TSF data, the Composite Software and TSF data as well as user data of the Composite Product. The protection covers reverse engineering, manipulation of hardware and security services as well as undetected modification of TOE memory content.

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.Abuse-Func is countered by O.Abuse-Func, because the objective requires to prevent the abuse of TOE functions which are disabled before TOE Delivery. The considered abuse covers disclosure or manipulation of critical TSF data or user data of the Composite Product as well as manipulation of Composite Software and bypass, deactivation, change or exploitation of security features or security services of the TOE, including test and debug functionality.

T.RND is countered by O.RND, because the objective requires detection and/or prevent manipulation or influence of the entropy source to ensure cryptographic quality of random number generation.

T.Insecure-State is countered by O.Secure-State, because the objective requires a secure initialisation 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.

The justification related to the organisational security policy “Identification of each TOE instance (P.Gen-Unique-ID)” is as follows:

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 organisational measures are concerned.

The justification related to the assumption “Treatment of user data of the Composite TOE (A.Resp-Appl)” is as follows:

OE.Resp-Appl requires the Composite Software to implement measures as assumed in A.Resp-Appl, so the assumption is covered by the objective.

The justification related to the assumption “Protection during Packaging, Finishing and Personalisation (A.Process-Sec-IC)” is as follows:

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.

The justification related to the assumption Packaging of the TOE (OE.Packaging-Requirement)” is as follows:

OE.Packaging-Requirement requires that the 3S developer provides a specification for the stacking, assembly and packaging, so this guidance can be followed as assumed in A.Packaging-Requirement, and the assumption is covered by this objective.

5 Extended Components Definition

The definition of the IT security functionality of the 3S requires additional SFRs that are not defined in Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components.

5.1 Definition of the Family FCS_RNG

An additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined for the random number generator. This family describes the functional requirements for random number generation used for cryptographic purposes.

FCS_RNG **Generation of random numbers**

Family behaviour:

This family defines quality requirements for the generation of random numbers which are intended to be use for cryptographic purposes.

Component levelling:

FCS_RNG Generation of random numbers	1

FCS_RNG.1 Generation of random numbers requires that random numbers meet a defined quality metric.

Management: FCS_RNG.1

There are no management activities foreseen.

Audit: FCS_RNG.1

There are no actions defined to be auditable.

FCS_RNG.1 **Random number generation**

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*] random number generator that implements: [assignment: *list of security capabilities*].

FCS_RNG.1.2 The TSF shall provide [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*] that meet [assignment: *a defined quality metric*].

Application Note 24. A physical random number generator (RNG) produces the random number by a noise source, based on physical random processes. A non-physical true RNG uses a noise source based on non-physical random processes, such as human interaction (key strokes, mouse movement). A deterministic RNG uses a random seed to produce a pseudorandom output. A hybrid RNG combines the principles of physical and deterministic RNGs where a physical RNG produces at least the amount of entropy the RNG output may contain, and the internal state of a

deterministic RNG output contains fresh entropy but less than the output of RNG may contain.

5.2 Definition of the Family FMT_LIM

The additional family (FMT_LIM) of the Class FMT (Security Management) describes the functional requirements for Test and/or Debug Features of the TOE. The new functional requirements were defined in the class FMT because this class addresses the management of functions of the TSF. The TOE implements technical mechanisms to address the specific issues of preventing the abuse of functions by limiting the capabilities of the functions and by limiting their availability. The functional requirement allows a combination of technical mechanisms to limit the capabilities and the availability. Therefore, the definition includes a dependency between the two components.

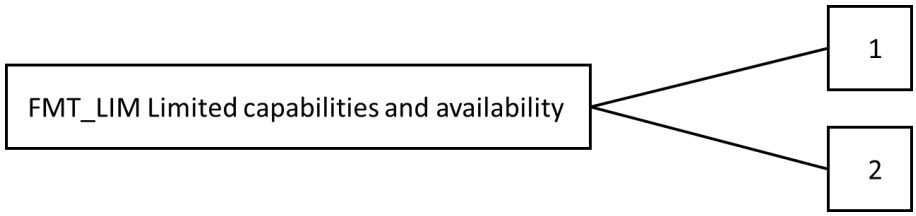
The family "Limited capabilities and availability (FMT_LIM)" is specified as follows.

FMT_LIM Limited capabilities and availability

Family behaviour:

This family defines requirements that limit the capabilities and availability of functions in a combined manner. Note that the family FDP_ACF restricts the access to functions whereas the component Limited Capability of this family requires the functions themselves to be designed in a specific manner.

Component levelling:



FMT_LIM.1 Limited capabilities requires that the TSF is built to provide only the capabilities (perform action, gather information) necessary for its genuine purpose.

FMT_LIM.2 Limited availability requires that the TSF restrict the use of functions (see Limited Capabilities (FMT_LIM.1)). This can be achieved by removing or by disabling functions in a specific phase of the TOE's life-cycle, for example.

Management: FMT_LIM.1, FMT_LIM.2

There are no management activities foreseen.

Audit: FMT_LIM.1, FMT_LIM.2

There are no actions defined to be auditable.

The TOE Functional Requirement "Limited capabilities (FMT_LIM.1)" is specified as follows.

FMT_LIM.1 Limited capabilities

Hierarchical to: No other components.

Dependencies: FMT_LIM.2 Limited availability.

FMT_LIM.1.1 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 [assignment: *Limited capability policy*].

The TOE Functional Requirement "Limited availability (FMT_LIM.2)" is specified as follows.

FMT_LIM.2 Limited availability

Hierarchical to: No other components.

Dependencies: FMT_LIM.1 Limited capabilities.

FMT_LIM.2.1 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 [assignment: *Limited availability policy*].

Application Note 25. The functional requirements FMT_LIM.1 and FMT_LIM.2 assume that there are two types of mechanisms (limitation of capabilities and limitation of availability) which together shall provide protection in order to enforce the same policy or two mutual supportive policies related to the same functionality. E.g., this enables the following:
(i) The TSF is provided without restrictions in the product in its user environment but its capabilities are so limited that the policy is enforced.
Or, conversely:
(ii) The TSF is designed with high functionality, but is removed or disabled in the product in its user environment.

5.3 Definition of the Family FAU_SAS

The additional family (FAU_SAS) of the Class FAU (Security Audit) is defined to describe the functional requirements for the storage of audit data. It has a more general approach than FAU_GEN, because it does not necessarily require the data to be generated by the TOE itself and because it does not give specific details of the content of the audit records.

The family "Audit data storage (FAU_SAS)" is specified as follows.

FAU_SAS Audit data storage

Family behaviour:

This family defines functional requirements for the storage of audit data.

Component levelling:

FAU_SAS.1 Requires the TOE to provide the possibility to store audit data.

Management: FAU_SAS.1

There are no management activities foreseen.

Audit: FAU_SAS.1

There are no actions defined to be auditable.

FAU_SAS.1 Audit storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FAU_SAS.1.1 The TSF shall provide [assignment: *list of subjects*] with the capability to store [assignment: *list of audit information*] in the [assignment: *type of persistent memory*].

5.4 Definition of the Family FDP_SDC

The Protection Profile defines the additional family (FDP_SDC.1) of the Class FDP (User data protection) to address confidentiality requirements for user data while stored under control of the TSF. The existing SFR on user data is limited to integrity protection.

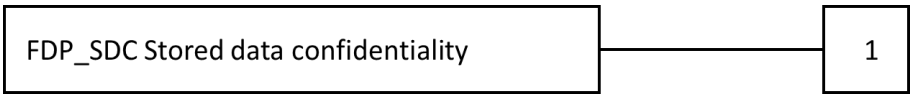
The family "Stored data confidentiality (FDP_SDC)" is specified as follows.

FDP_SDC Stored data confidentiality

Family behaviour:

This family provides requirements that address protection of user data confidentiality while these data are stored within memory areas protected by the TSF. The TSF provides access to the data in the memory through the specified interfaces only and prevents compromising their information bypassing these interfaces. It complements the family "Stored data integrity (FDP_SDI)" which protects the user data from integrity errors while being stored in the memory.

Component levelling



FDP_SDC.1 Requires the TOE to protect the confidentiality of information of the user data in specified memory areas.

Management: There are no management activities foreseen.

Audit: There are no actions defined to be auditable.

FDP_SDC.1 Stored data confidentiality

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SDC.1.1 The TSF shall ensure the confidentiality of the information of the user data while it is stored in the [assignment: *memory area*].

5.5 Definition of the Family FPT_INI

The additional family (FPT_INI) of the Class FPT (Protection of the TSF) is defined to describe the functional requirements for the secure initialisation of the TSF. This family describes the functional requirements for the initialisation of the TSF by a dedicated security functionality of the TOE that ensures the initialisation in a correct and secure operational state.

The family "TSF Initialisation (FPT_INI)" is specified as follows.

FPT_INI **TSF Initialisation**

Family behaviour:

This family defines functional requirements for the secure initialisation of the TSF.

Component levelling:

FPT_INI.1 Requires the TOE to enforce a secure initialisation of the TSF.

Management: FPT_INI.1

There are no management activities foreseen.

Audit: FPT_INI.1

There are no actions defined to be auditable.

FPT_INI.1 **TSF Initialisation**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_INI.1.1 The TOE initialization function shall verify [assignment: *list of verifications*] 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.

5.6 Definition of the Family FPT_EMS

The Protection Profile defines the additional family (FPT_EMS.1) of the Class FPT (Protection of the TSF) to describe the protection of the TOE against leakage of TSF data and user data while being stored or processed by the TOE.

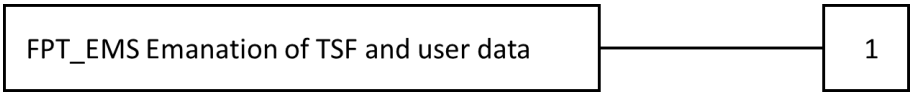
The family "Emanation of TSF and User data (FPT_EMS)" is specified as follows.

FPT_EMS Emanation of TSF and user data

Family behaviour:

This family requires that leakage of the TOE cannot be used to compromise sensitive TSF data or user data. The leakage may occur when TSF data is transferred or processed by the TOE hardware.

Component levelling



FPT_EMS.1 Requires the TOE to protect TSF data and user data against leakage that may be generated during transfer or processing of such data inside the TOE.

Management: There are no management activities foreseen.

Audit: There are no actions defined to be auditable.

FPT_EMS.1 Emanation of TSF and user data

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_EMS.1.1 The TSF shall ensure that the TOE does not emit emissions over its attack surface to such an extent that these emissions enable access to TSF data and user data, as specified in the following table:

ID	Emanation	Attack Surface	TSF data	User Data
1	[assignment: *list of types of emissions*]	[assignment: *list of types of attack surface*]	[assignment: *list of types of TSF data*]	[assignment: *list of types of user data*]

Table 3: Definition of Side-Channel Protection

6 IT Security Requirements

6.1 Security Functional Requirements for the TOE

The operations of the Security Functional Requirements (SFRs) are identified in the following way:

The refinement operation is used to add detail to a requirement, and, therefore, further restricts a requirement. Refinements of security requirements are denoted in such a way that added words are in bold text and removed words are crossed out. In some cases, an interpretation refinement is given. In such cases, an extra paragraph starting with "Refinement" provides the related rationale.

The selection operation is used to select one or more options provided by the CC in stating a requirement. Selections made by the PP author are denoted as underlined text. Selections to be filled in by the ST author appear in square brackets with an indication that a selection is to be made [selection:] and are italicised.

The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments made by the PP author are denoted as underlined text. Assignments to be filled in by the ST author appear in square brackets with an indication that an assignment is to be made [assignment:] and are italicised. In some cases, the assignment made by the PP authors defines a selection to be performed by the ST author. Therefore, this text is underlined and italicised.

The iteration operation is used when a component is repeated with varying operations. Iteration is denoted by showing a forward slash "/", and the iteration indicator after the component identifier.

6.1.1 Protection against Malfunction

The TOE shall either tolerate disturbance (e.g., from external operating conditions) or, if malfunctions cannot be prevented, stop the operations. The TOE shall be protected from misconfiguration and by-passing by means of the Composite Software. These aspects are addressed by the security assurance requirements Architectural design (ADV_ARC.1).

The TOE shall meet the requirement "Limited fault tolerance (FRU_FLT.2)" as specified below.

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	The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: exposure to operating conditions or usage conditions out of range, which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1) or abnormal interface behaviour and/or protocol parameters or protocol sequences that can be tolerated because the correct operation of the TSF is ensured[4].
Refinement:	**The term "failure" above means "circumstances". The TOE prevents failures for the "circumstances" defined above.**

Application Note 26. Environmental conditions include but are not limited to power supply, clock, and other external signals (e.g., a reset signal) necessary for the TOE operation.

The TOE shall meet the requirement "Failure with preservation of secure state (FPT_FLS.1)" as specified below.

[4] [assignment: *list of types of failures*]

FPT_FLS.1	**Failure with preservation of secure state**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: exposure to operating conditions or usage conditions with respect to interfaces and communication which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2) and where, therefore, a malfunction could occur[5].
Refinement:	**The term "failure" above also covers "circumstances". The TOE prevents failures for the "circumstances" defined above.**

Application Note 27. The Security Target shall describe the secure state. In addition, the author of the Security Target should give a rationale together with a clear definition of the secure state here.

Application Note 28. The Common Criteria suggest that the TOE generates audit data for the SFRs Limited fault tolerance (FRU_FLT.2) and Failure with preservation of secure state (FPT_FLS.1). This may be advantageous or even required for the application context. The author of the Security Target should consider this especially for FPT_FLS.1.

6.1.2 Protection against Abuse of Functionality

The 3S may implement test functions to support the functional testing after the production. The TOE shall prevent abuse of such functionality after the test phase. The protection can be achieved either by limiting the capability of the implemented functions or limiting the availability. Limited capability prevents misuse or compromise of TSF data or user data, or the characterisation of security functions and security services, even if the function can be reactivated, while limited availability prevents access to the functionality after testing. In most cases, both types of limitations are implemented to ensure the required protection.

The 3S may provide debugging services based on specific configuration of the TOE. The TOE prevents the use of this debugging functionality to prevent misuse or compromise of TSF data or user data, or perform characterisation of security functions and security services. The debugging functionality may be limited, however, in terms of its capabilities and availability.

Test functionality and debug functionality may be limited by independent security mechanisms, so the SFRs defining the associated protection are iterated.

The TOE shall meet the requirement "Limited capabilities (FMT_LIM.1)" to prevent the misuse of test functionality, as follows:

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 and implemented 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 of the Composite TOE to be disclosed or manipulated, TSF data to be

[5] [assignment: *list of types of failures in the TSF*]

	disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks[6].

The TOE shall meet the requirement "Limited availability (FMT_LIM.2)" as specified below to prevent the misuse of test functionality.

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 and implemented 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 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].

The TOE shall meet the requirement "Limited capabilities (FMT_LIM.1)" as specified to prevent the misuse of debug functionality.

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 and implemented 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[8].

The TOE shall meet the requirement "Limited availability (FMT_LIM.2)" as specified below to prevent the misuse of debug functionality.

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 and implemented 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 of the Composite TOE to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no

[6] [assignment: *Limited capability and availability policy*]
[7] [assignment: *Limited capability and availability policy*]
[8] [assignment: *Limited capability and availability policy*]

substantial information about construction of TSF to be gathered which may enable other attacks[9].

6.1.3 Protection against Physical Manipulation and Probing

The TOE shall meet the requirement "Stored data confidentiality (FDP_SDC.1/3S)" as specified below.

FDP_SDC.1/3S	**Stored data confidentiality**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SDC.1.1/3S	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the [assignment: *memory area*].

The TOE shall meet the requirement "Stored data integrity monitoring and action (FDP_SDI.2/3S)" as specified below.

FDP_SDI.2/3S	**Stored data integrity monitoring and action**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1/3S	The TSF shall monitor user data stored in containers controlled by the TSF for [assignment: *integrity errors*] on all objects, based on the following attributes: [assignment: *user data attributes*].
FDP_SDI.2.2/3S	Upon detection of a data integrity error, the TSF shall [assignment: *action to be taken*].

Application Note 29. The Security Target writer shall perform the open operations. It may assign the monitored memory areas as user attributes in the element FDP_SDI.2.1.

The TOE shall meet the requirement "Resistance to physical attack (FPT_PHP.3)" as specified below.

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[10] to the TSF[11] 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, to ensure that SFRs are enforced. Therefore, in this case, "automatic response" means (i)**

9 [assignment: *Limited capability and availability policy*]
10 [assignment: *physical tampering scenarios*]
11 [assignment: *list of TSF devices/elements*]

assuming that there might be an attack at any time and (ii) countermeasures are provided at any time.

Application Note 30. The Security Target shall describe the automatic response of the TOE. All SFRs are derived from security objectives to protect the user data and TSF data stored and processed by the 3S, or to provide secure security services. Therefore, the SFRs are enforced if the TOE stops operation or does not operate at all if a physical manipulation or physical probing attack is detected and the security cannot be ensured in another way.

6.1.4 Protection against Leakage

FPT_EMS.1	**Emanation of TSF and user data**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_EMS.1.1	The TSF shall ensure that the TOE does not emit emissions over its attack surface to such an extent that these emissions enable access to TSF data and user data, as specified in the following table:

ID	Emanation	Attack Surface	TSF data	User Data
1	Power	Power distribution layer of the SoC and interfaces of the 3S and SoC	Private components of keys and generation of random numbers, [assignment: list of types of TSF data]	Private components of keys, generation of random numbers and other secrets such as PIN or password, [assignment: list of types of user data]
2	Emission	Both sides (substate and wire routing) of the complete SoC chip surface	Private components of keys and generation of random numbers [assignment: list of types of TSF data]	Private components of keys, generation of random numbers and other secrets such as PIN or password, [assignment: list of types of user data]
3	Timing	Power distribution layer of the SoC and interfaces of the 3S and SoC as well as both sides (substrate and wire routing) of the complete SoC chip surface	Private components of keys and generation of random numbers [assignment: list of types of TSF data]	Private components of keys, generation of random numbers and other secrets such as PIN or password, [assignment: list of types of user data]
4	[assignment: list of types of emissions]	[assignment: list of types of attack surface]	[assignment: list of types of TSF data]	[assignment: list of types of user data]

Table 4: Definition of Side-Channel Protection

Application Note 31. Depending on the implementation of the 3S and the design of the SoC the ST author may extend the table above.

6.1.5 TOE Identification and Root of Trust

The TOE shall meet the requirement "Audit storage (FAU_SAS.1)" as specified below (Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components extended).

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[12] with the capability to store [selection: TOE unique identification data, *Initialisation Data, Pre-personalisation Data, [assignment: other data]*][13] in the [assignment: *type of persistent memory*].

Application Note 32. The integrity and uniqueness of the unique identification of the TOE shall be supported by the development, production and test environment.

Application Note 33. The ST writer shall perform the operation in the element FAU_SAS.1.1 by selecting/assigning the type of data and by assigning the type of persistent memory provided for the storage of Initialisation Data and/or Pre-personalisation Data. If the TOE provides specific functions to protect these data or to process them, appropriate SFRs can be specified in the ST. Then the above paragraph needs to be revised accordingly.

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[14], *[assignment: list of verifications]* 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.

6.1.6 Generation of Random Numbers

The TOE generates random numbers. To define the IT SFRs of the TOE an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined in section 5.1. This family FCS_RNG Generation of random numbers describes the functional requirements for random number generation used for cryptographic purposes.

---

12 [assignment: *list of subjects*]

13 [assignment: *list of audit information*]

14 [assignment: list of verifications]

The TOE shall meet the requirement "Quality metric for random numbers (FCS_RNG.1)" as specified below (Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components extended).

FCS_RNG.1	**Random number generation**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1	The TSF shall provide a [selection: *physical, hybrid physical, hybrid deterministic*][15] random number generator that implements: [assignment: *list of security capabilities*].
FCS_RNG.1.2	The TSF shall provide [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*] that meet [assignment: *a defined quality metric*].

Application Note 34. The ST writer shall perform the open operations. The operation performed in the element FCS_RNG.1.1 selects RNG types based on physical random number generators, as typically provided by 3S. Chapter 9.2 provides examples for the security capabilities and quality metrics used in some national certification schemes.

6.2 Security Assurance Requirements for the TOE

The Security Target to be developed based upon this Protection Profile will be evaluated according to Security Target evaluation (Class ASE).

The Security Assurance Requirements for the evaluation of the TOE are those taken from the

- Evaluation Assurance Level 4 (EAL4)

and augmented by taking the following components:

- ALC_DVS.2, ATE_DPT.2 and AVA_VAN.5.

The assurance requirements are:

Class ADV: Development

Architectural design	(ADV_ARC.1)

Application Note 35. Refinements related to the integration guidance:
- ADV_ARC.1.1D: The developer shall provide a rationale for the correct integration of the 3S in the SoC as part of the TSF security architecture description.
- ADV_ARC.1.1C: The rationale shall be at the level of detail of the TOE design and the integration guidance requirements.
- ADV_ARC.1.1E in CEM: TBD

Functional specification	(ADV_FSP.4)
Implementation representation	(ADV_IMP.1)
TOE design	(ADV_TDS.3)

[15] [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]

Class AGD: Guidance documents

Operational user guidance	(AGD_OPE.1)
Preparative user guidance	(AGD_PRE.1)

Application Note 36. The SoC integrator should be identified as a User. Therefore, integration guidance shall be evaluated as part of the AGD class.

Class ALC: Life-cycle support

CM capabilities	(ALC_CMC.4)
CM scope	(ALC_CMS.4)
Delivery	(ALC_DEL.1)
Development security	(ALC_DVS.2)
Life-cycle definition	(ALC_LCD.1)
Tools and techniques	(ALC_TAT.1)

Class ASE: Security Target evaluation

Conformance claims	(ASE_CCL.1)
Extended components definition	(ASE_ECD.1)
ST introduction	(ASE_INT.1)
Security objectives	(ASE_OBJ.2)
Derived security requirements	(ASE_REQ.2)
Security problem definition	(ASE_SPD.1)
TOE summary specification	(ASE_TSS.1)

Class ATE: Tests

Coverage	(ATE_COV.2)
Depth	(ATE_DPT.2)
Functional tests	(ATE_FUN.1)
Independent testing	(ATE_IND.2)

Class AVA: Vulnerability assessment

Vulnerability analysis	(AVA_VAN.5)

Application Note 37. This Protection Profile requires EAL4 augmented but allows higher hierarchical components to be added. To support this, most parts of the Protection Profile are - whenever possible - formulated independently from possible augmentations (e.g., those to reach EAL5 augmented). Therefore, this Protection Profile often refers to “the Common Criteria assurance component of the family XY” instead of referring to the specific components listed above. If the Security Target uses further augmentations this shall be identified in this section. The authors of the Security Target shall also review the rationale of this Protection Profile and extend it as appropriate.

6.3 Security Requirements Rationale

6.3.1 Rationale for the SFRs

Table 5 provides an overview, how the SFRs are combined to meet the security objectives. The justification for each objective is detailed after the table.

Objective	TOE Security Functional and Assurance Requirements
O.Leak-Inherent	FPT_EMS.1 Emanation of TSF and user data
O.Phys-Probing	FDP_SDC.1/3S Stored data confidentiality FPT_PHP.3 Resistance to physical attack
O.Malfunction	FRU_FLT.2 Limited fault tolerance FPT_FLS.1 Failure with preservation of secure state Supported by: FPT_INI.1 TSF Initialisation
O.Phys-Manipulation	FDP_SDI.2/3S Stored data integrity monitoring and action FPT_PHP.3 Resistance to physical attack
O.Leak-Forced	FPT_EMS.1 Emanation of TSF and user data FRU_FLT.2 Limited fault tolerance FPT_FLS.1 Failure with preservation of secure state 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 Test Limited capabilities FMT_LIM.2/Debug Limited availability Supported by: FAU_SAS.1 Audit storage FRU_FLT.2 Limited fault tolerance FPT_FLS.1 Failure with preservation of secure state FDP_SDI.2/3S Stored data integrity monitoring and action FPT_PHP.3 Resistance to physical attack
O.RND	FCS_RNG.1 Random number generation Supported by: FPT_FLS.1 Failure with preservation of secure state FPT_FLT.2 Limited fault tolerance FPT_EMS.1 Emanation of TSF and user data FPT_PHP.3 Resistance to physical attack
O.Secure-State	FPT_INI.1 TSF Initialisation Supported by: FRU_FLT.2 Limited fault tolerance FPT_FLS.1 Failure with preservation of secure state FDP_SDI.2/3S Stored data integrity monitoring and action FPT_PHP.3 Resistance to physical attack
O.Identification	FAU_SAS.1 Audit storage Supported by: FPT_INI.1 TSF Initialisation

Table 5: Security Requirements versus Security Objectives

The justification related to the security objective “Protection against Inherent Information Leakage (O.Leak-Inherent)” is as follows:

The SFR FPT_EMS.1 explicitly requires the prevention of emission that enables access to secret data (TSF data as well as user data) over the TOE attack surface. According to the already performed assignment, this covers power, emanation and timing. The attack surface comprises the chip surface as well as all interfaces of the 3S.

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 shall be addressed in the Guidance Documentation. FPT_EMS.1 together with the guidance are suitable to meet the objective

The justification related to the security objective “Protection against Physical Probing (O.Phys-Probing)” is as follows:

The SFR FDP_SDC.1 requires the TSF to protect the confidentiality of the information of user data and TSF data stored in specified memory areas and prevent their compromising 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, this SFR supports the objective.

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

The justification related to the security objective “Protection against Malfunctions (O.Malfunction)” is as follows:

The definition of this objective shows that it covers a situation where malfunction of the TOE might be caused by the operating conditions of the TOE (while direct manipulation of the TOE is covered by O.Phys-Manipulation). The security objective covers the following two aspects according to operating conditions: 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, because it states that a secure state is preserved in this case. The first case is covered by FRU_FLT.2, because it states that the TOE operates correctly under normal (tolerated) conditions. The same applies also for abnormal interface behaviour and/or protocol parameters or protocol sequences that do not meet the specified behaviour. The TOE enters a secure state covered by FPT_FLS.1 in case the circumstances lead to an insecure operation. The TOE continues operation as long as the abnormal behaviour or protocol failures do not impact the intended processing of the TOE covered by FRU_FLT.2.

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 and FPT_FLS.1 shall work independently so that their operation cannot be affected by the Composite Software (see the refinement). Therefore, there is no possible instance of conditions under O.Malfunction, which is not covered.

The justification related to the security objective “Protection against Physical Manipulation (O.Phys-Manipulation)” is as follows:

The SFR FDP_SDI.2/3S defines a security mechanism to detect integrity errors of the stored user data and TSF data and react to 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 SFR supports the objective.

It is possible that the TOE needs additional support by the FW, SW and Composite Software (e.g., by implementing FDP_SDI.2) to check data integrity with the help of appropriate checksums. This support shall be addressed in the Guidance Documentation. Together with FPT_PHP.3, this is suitable to meet the objective.

The justification related to the security objective “Protection against Forced Information Leakage (O.Leak-Forced)“ is as follows:

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 behaviour of the TOE (either by exposing it to extreme operating conditions or by directly manipulating it) with a second attack step measuring and analysing some output produced by the TOE. The first step is prevented by the SFR FRU_FLT.2 and FPT_FLS.1 for the control of the operating conditions and FPT_PHP.3 that prevent physical modification. Furthermore, the protection against leakage defined by FPT_EMS.1 supports O.Leak-Forced, because it prevents the attacker from being successful if he tries the second step directly (e.g., with operating conditions at their limits that are not detected).

The justification related to the security objective “Protection against Abuse of Functionality (O.Abuse-Func)” is as follows:

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 7 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 7 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 and FPT_FLS.1 control the operating conditions and prevent malfunctions that may allow to circumvent the control implemented by FMT_LIM.1 and FMT_LIM.2. The SFR FDP_SDI.2/3S 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. The supporting SFR overview is included in Table 5.

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.

The justification related to the security objective “Random Numbers (O.RND)” is as follows:

FCS_RNG.1 requires the TOE to provide random numbers of good quality. The specification of the exact metric is left to the individual Security Target for a specific TOE.

The SFRs FPT_FLS.1 and FPT_FLT.2 prevent malfunction of the TOE, based on malicious operating conditions. FPT_PHP.3 prevents physical manipulation and FPT_EMS.1 prevents leakage that may disclose data generated by the random number generator.

Random numbers are mainly used by the Composite Software to generate cryptographic keys for internal use. Therefore, the TOE shall prevent the unauthorised disclosure of random numbers. Other SFRs, which support the prevention of inherent leakage attacks, probing and forced leakage attacks, ensure the confidentiality of the random numbers provided by the TOE.

The FW, SW or the Composite Software have to support the objective by providing runtime-tests of the random number generator, depending on the implementation of the random number generator and the associated protection in a specific TOE. Together, these requirements allow the TOE to provide random numbers with high entropy 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, because Part 2 of the Common Criteria does not contain generic SFRs for Random Number generation.

The justification related to the security objective “Secure start-up and re-start (O.Secure-State)” is as follows:

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 start-up 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 and FPT_FLS.1 controlling the operating conditions and preventing malfunctions that may allow to manipulate the secure initialisation. The SFR FDP_SDI.2/3S 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. The supporting SFR overview is included in Table 5.

The justification related to the security objective “TOE Identification (O.Identification)“ is as follows:

This objective states that the TOE shall be able to provide a unique identification of the TOE instance. The SFR defines the capability to store audit information provided by a subject in a persistent memory of the TOE. Therefore, the SFRs are suitable to meet the objective.

O.Secure-State requires the correct initialisation and configuration of the TOE. This includes the integrity check of the unique identifier of the TOE. Therefore, this objective supports O.Identification.

6.3.2 Dependencies of SFRs

Table 6 lists the SFRs defined in this Protection Profile, their dependencies and whether they are satisfied by other security requirements defined in this Protection Profile. The text following the table discusses the remaining cases

Requirement	Dependency	Satisfied Dependency
FPT_EMS.1	None	No dependency

Requirement	Dependency	Satisfied Dependency
FPT_PHP.3	None	No dependency
FDP_SDC.1/3S	None	No dependency
FRU_FLT.2	FPT_FLS.1	Satisfied by FPT_FLS.1
FPT_FLS.1	No dependency	No dependency
FDP_SDI.2/3S	No dependency	No dependency
FMT_LIM.1/Test	FMT_LIM.2	Satisfied by FMT_LIM.2/Test
FMT_LIM.2/Test	FMT_LIM.1	Satisfied by FMT_LIM.1/Test
FMT_LIM.1/Debug	FMT_LIM.2	Satisfied by FMT_LIM.2/Debug
FMT_LIM.2/ Debug	FMT_LIM.1	Satisfied by FMT_LIM.1/Debug
FCS_RNG.1	None	No dependency
FPT_INI.1	None	No dependency
FAU_SAS.1	None	No dependency

Table 6: Overview of SFR dependencies

6.3.3 Rationale for the Assurance Requirements

The assurance level EAL4 and the augmentation with the requirements ALC_DVS.2, ATE_DPT.2 and AVA_VAN.5 were chosen in order to meet assurance expectations explained in the following paragraphs.

An assurance level of EAL4 with the augmentations ATE_DPT.2, AVA_VAN.5 and ALC_DVS.2 is required for this type of TOE, because 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 defence against such attacks, the evaluators should have access to a sufficiently detailed TOE Design Specification and the source code.

6.3.3.1 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.

In the particular case of a 3S hardware design block, the TOE is developed and produced within a complex and distributed industrial process which shall be protected in particular. Details about the implementation, (e.g., from design, test and development tools as well as Initialisation Data) may make such attacks easier. Therefore, in the case of a hardware design block, maintaining the confidentiality of the design is very important. ALC_DVS.2 includes requirements to continuously assess the security measures and verify the applicability and sufficiency for all sensitive configurations items that are part of the TOE.

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

6.3.3.2 ATE_DPT.2 Advanced methodical vulnerability analysis

The selection of the component ATE_DPT.2 provides a higher assurance by requiring the functional testing of SFR-enforcing modules. The TOE provides a hardware platform where the more comprehensive test analysis supports the resilient functionality of security features and security services.

ATE_DPT.2 has dependencies to ADV_ARC.1 “Security architecture description”, ADV_FSP.2 “Security enforcing functional specification”, ADV_TDS.3 “Basic modular design”, and ATE_FUN.1 “Functional testing”.

All these dependencies are satisfied by EAL4.

6.3.3.3 AVA_VAN.5 Advanced methodical vulnerability analysis

Due to the intended use of the TOE, it shall 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 try to attack 3Ss, such as the TOE used for payment systems, Subscriber Identity Module (SIM), storage and management of digital identities. Therefore, AVA_VAN.5 was chosen specifically to assure that even these attackers cannot successfully attack the TOE.

7 Definition of Packages

The following packages can be added to the base Protection Profile. Each package defines an extension of the TOE functionality.

Some of the packages have dependencies that need to be considered, see section 1.3.

7.1 Package for Passive External Memory

This package describes the extension of the security problem definition and the SFRs, if the 3S is connected to passive external memory. The passive external memory does not provide any security functionality and is outside the boundary of the TOE. The usage of passive memory outside the TOE has the following effects:

- The TOE implements an interface to the internal SoC bus to access the passive external memory. The SoC implements the interface to the external memory that is shared by the SoC and the 3S. The passive external memory does not implement any security service or security functionality, so the external memory is named passive external memory.
- Passive external memory can store an encrypted and authenticated software image that can either be loaded in the TOE during start-up or during runtime. In this case the TOE implements a security service to authenticate, verify the integrity and decrypt the content of the software image before it is executed in the TOE. Further on, the security service prevents rollback to older versions of the software image. When TOE FW/SW is activated, the TOE can load Composite Software to be executed by the TOE as user data.
- The passive external memory can also store a firmware image to enable updates of the firmware. Loading Firmware images require a similar security service than the loading of software images.
- Further on, the TOE can store TSF data and User Data in the passive external memory as protected data container. The security functionality for TSF data and User Data shall enforce confidentiality, integrity, freshness and replay protection.

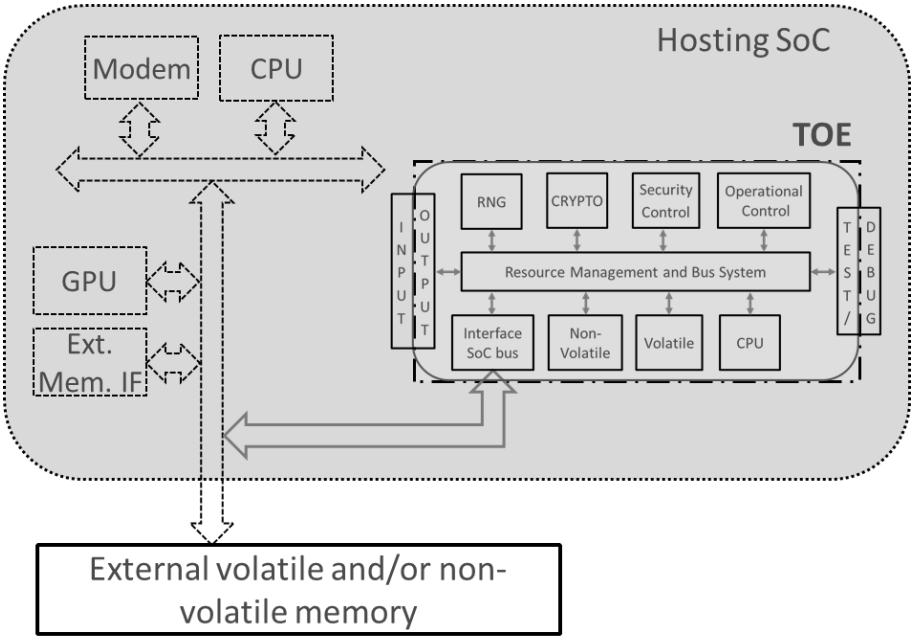


Figure 6: 3S with passive external memory (PM)

Attacks on data stored in passive external memory shall be detected to protect the TOE against the consequences of such attacks outside the TOE boundary, because the passive external memory is

shared with the remaining components of the SoC. Therefore, additional threats shall be included in the Security Target.

7.1.1 Security Problem Definition

7.1.1.1 Description of Assets

Application Note 38. There are no additional assets defined in this package.

7.1.1.2 Threats

The following figure describes the attacks on the TOE with passive external memory. The threats described in this section shall be added in the Security Target together with the threats against the TOE described for the base configuration (see section 3.2).

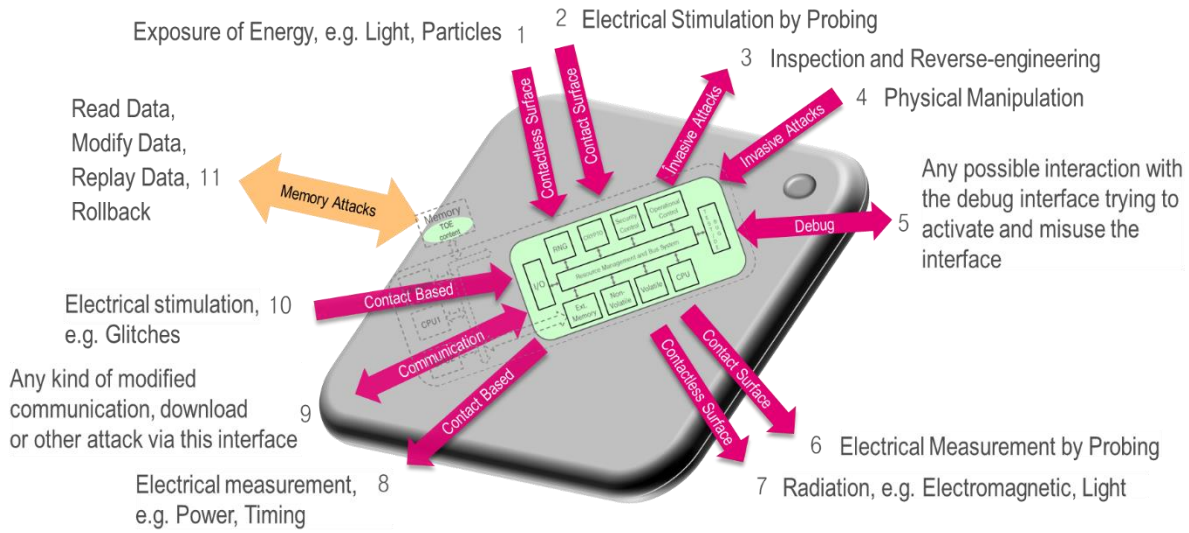


Figure 7: Attacks against passive external memory

Grey box represents the SoC with TOE (green box) and its interaction channels. The external memory may store a protected software image and data that both belong to the TOE.

The TOE shall protect against the threat “Cloning the TOE with a copy of the passive external memory (T.Pas-Mem-Clone-Replace)” as specified below.

T.Pas-Mem-Clone-Replace	Cloning or replacement of passive external memory 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.

This threat refers to the case where partial or full content of the external memory is cloned to a different device. It can also cover the replacement of the physical external memory used by the 3S with the memory of a different unit. The second case might not be viable on some architectures or memory when the physical design or assembly procedures impede it.

The effect of this threat is in replacing the data and/or image of a TOE with a different one and to obtain a valid but unauthorised instance of the TOE.

This threat involves using two different TOE units or instances. One TOE unit is used as a source for the external memory content. This content is used to replace the genuine content of the external memory of the second TOE unit.

Another possible scenario for this threat can be contemplated for passive external non-volatile memory: the external non-volatile memory is replaced with an empty or virgin non-volatile memory, removing the user and TSF data used by the TOE, and possibly forcing the TSF to generate new user and TSF data, potentially affecting the TSF behaviour.

The TOE shall protect against the threat “Abuse of passive external memory content (T.Pas-Mem-Content-Abuse)” as specified below.

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.

An attacker may obtain unauthorised access to the external memory and attempt to read, disclose, modify or replace the content of the external memory. This threat addresses also the authenticity of the data stored in the external memory.

Note that the access to the external memory or the transfer of data between the TOE and the external memory may also support an attack.

The TOE shall avert the threat “Replay of commands between the 3S and the passive external memory (T.Pas-Mem-Cmd-Replay)” as specified below.

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.

The read, write and erase commands issued by the 3S to exercise the storage functionality of the external memory, and their payloads, can be intercepted by an attacker (e.g., eavesdrop the commands on the link between the 3S and the external memory). Such an attacker may use copies of these commands to try to misuse the TOE or compromise data. The command replay attack can take the following forms:

- The attacker reacts to a read command and replies with a previously recorded answer (e.g., to a previous read request). In this way, the 3S gets an old version of such content.
- The attacker issues a previous write command, trying to overwrite the external memory with the previous content, and leading to the 3S obtaining old versions of such content in later read operations.
- The attacker issues a previous erase command, trying to overwrite status information or other data that may lead to misuse of the TOE.

The TOE shall avert the threat “Unauthorised rollback of content in the passive external memory (T.Pas-Mem-Unauth-Rollback)” as specified below.

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.

This threat takes advantage of the fact that the external memory is not integrated into the 3S. Hence, physical protections for preventing the replacement of content may not cover the external memory. This situation enables an attacker to read and write the content of the external memory. Even if the confidentiality and integrity of the external memory content is protected, the replacement with an old copy may also be valid, because it is retrieved from the external memory.

If the TOE image is stored in an external memory, this threat may lead to an unauthorised rollback of the TOE image to an older version. Even when the TOE stores data and not code in the external memory, this data rollback might affect the behaviour of the TSF.

The replacement of content stored in the external memory with previous versions of it may refer to the full content of the external memory or partial content of it, depending on the organization and protection of the data stored in the external memory.

7.1.1.3 Organisational Security Policies

Application Note 39. There are no additional Organisational Security Policies defined in this package.

7.1.1.4 Assumption

Application Note 40. This package does not define an additional assumption.

7.1.2 Security Objectives

7.1.2.1 Security Objectives for the TOE

The TOE shall provide “Protection against disclosure and undetected modification of passive external memory content (O.Pas-Mem-Content-Prot)” as specified below.

O.Pas-Mem-Content-Prot: Protection against disclosure and undetected modification of passive external memory content.

The content in the external memory shall be protected against disclosure and undetected modification, because an attacker can directly access the external memory.

This security objective requires protection of the content stored in external memory by the TOE. The protection prevents disclosure and identifies modifications of stored code and data that is not performed by the TOE.

The TOE shall provide “Protection against replay of commands to store or modify data in passive external memory to the 3S (O.Pas-Mem-Cmd-Replay-Prot)” as specified below.

O.Pas-Mem-Cmd-Replay-Prot: Protection against replay of commands to store or modify data in passive external memory to the 3S.

The TOE shall protect against replay of content during write, read and erase operations to the external memory by the 3S.

This security objective requires protection against replay of read, write and erase operations. This covers simple replay of previously recorded commands or memory content but also the replay of modified commands or memory content. The TOE shall be able to detect such attacks violating the TOE.

The TOE shall provide “Protection against an unauthorised rollback of external memory content (O.Pas-Mem-Unauth-Rollback-Prot)” as specified below.

O.Pas-Mem-Unauth-Rollback-Prot: Protection against an unauthorised rollback of external memory content.

The TOE shall protect against replacement of the external memory content with a previous version, even if it was valid in the past.

The security objective requires protection against the simulation of outdated memory content. Replacement of memory content with a previous version of the same content or the manipulations of write operations violate the freshness of the external memory content and shall be detected by the TOE.

The TOE shall provide “Passive external memory content Irreversibility Anchor (O.Pas-Mem-Irreversible-Anchor)” as specified below.

O.Pas-Mem-Irreversible-Anchor Passive external memory content Irreversibility Anchor

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.

The security objective requires the verification of freshness for data read from the external memory. Therefore, the 3S shall maintain a reference that represents the current content of the external memory. This reference needs to be updated with each authorised read and write operation to detect a violation of the data freshness. It should be maintained in any TOE operational state, including the standby and sleep states. In the case of non-volatile memory, the Irreversibility Anchor needs to be persistently saved between two boots.

The TOE shall provide “Protection against passive external memory cloning or replacement (O.Pas-Mem-Clone-Replace-Prot)” as specified below.

O.Pas-Mem-Clone-Replace-Prot: Protection against passive external memory cloning or replacement.

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.

The security objective requires protection against replacement of its external memory content with the external memory content of another instance of the TOE. The external memory content shall only be valid for the 3S that is initially linked to this external memory. The replacement of the external memory or the transfer of the content from a memory that is linked to another instance of the TOE shall be detected.

7.1.2.2 Security Objectives for the TOE Environment

Application Note 41. This package does not include additional Security Objectives for the TOE Environment.

7.1.2.3 Security Objectives Rationale

	O.Pas-Mem-Content-Prot	O.Pas-Mem-Cmd-Replay-Prot	O.Pas-Mem-Irreversible-Anchor	O.Pas-Mem-Unauth-Rollback-Prot	O.Pas-Mem-Clone-Replace-Prot
T.Pas-Mem-Content-Abuse	X				
T.Pas-Mem-Cmd-Replay		X	X		
T.Pas-Mem-Unauth-Rollback			X	X	
T.Pas-Mem-Clone-Replace					X

Table 7: Mapping between objectives and threats

In the following, the justification of the coverage of the threats by the security objectives is given.

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 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 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 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.

7.1.3 Extended Component Definition

7.1.3.1 Definition of the Family FDP_URC

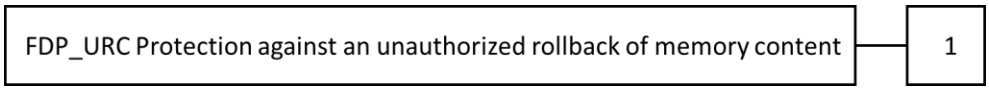
The Protection Profile defines the additional family (FDP_URC) of the Class FDP (User data protection) to verify the freshness of data stored in a physically separated memory. This family defines mechanisms to determine whether the content read from a physically separated memory meets the property of data freshness, by verifying that they are those resulting from the latest authorised operation (write or erase) of the TSF that modifies the content in the physically separated memory. If the content read from the physically separated memory cannot be uniquely linked to the latest write or erase operation executed by the TSF, the data freshness property is not met, and the read data is rejected.

FDP_URC: Protection against an unauthorised rollback of memory content

Family behaviour:

This family defines functional requirements for the detection of an unauthorised rollback of content stored in the external memory.

Component Levelling



FDP_URC.1 Requires the TOE to protect against an unauthorised rollback of the content stored in the external memory.

Management FDP_URC.1

There are no management activities foreseen.

Audit FDP_URC.1

There are no actions defined to be auditable.

FDP_URC.1 Protection against an unauthorised rollback of memory content

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_URC.1.1 The TOE shall detect an unauthorised replacement of the content stored in [assignment: *physically separated 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 Upon detection of unauthorised rollback of the content stored in a physically separated memory, the TOE shall [selection: *stop TOE operation, [assignment: other actions]*]

7.1.3.2 Definition of the Family FDP_IRA

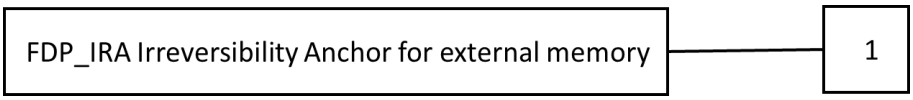
The family “Irreversibility Anchor of external memory content (FDP_IRA)” is specified as follows.

FDP_IRA Irreversibility Anchor for external memory

Family behaviour:

This family provides requirements for the implementation of a mechanism that verifies that read operations from this physically separated memory always represent the latest authorised modification of this memory. The TSF provides an Irreversibility Anchor that maintains a link between a transaction counter associated write or erase operation and the data transferred to a physically separated memory. Thereby, the Irreversibility Anchor allows to determine, whether a data read operation from the physically separated memory represents the data, based on the latest write or erase operation. The anchor is implemented in an irreversible way representing unique states (i.e., without the possibility of reverting to previous states). The pattern maintained by the Irreversibility Anchor value allows verification of the data freshness provided by subsequent read operations to the physically separated memory. If the physically separated memory is a non-volatile memory, the Irreversibility Anchor shall be maintained in any operational state of the TOE.

Component levelling



FDP_IRA.1 Requires the TOE to verify that read operations from a physically separated memory represent always the latest authorised modification of this memory.

Management: There are no management activities foreseen.

Audit: The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Any violation of the data freshness detected upon a read operation from the physically separated memory.

FDP_IRA.1 Irreversibility Anchor for external memory

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_IRA.1.1 The TSF shall verify the freshness of data for each read operation from the [assignment: physically separated memory].

FDP_IRA.1.2 The Irreversibility Anchor shall maintain a distinct transaction references for each [selection: *write, erase, [assignment: further operation that changes the content of the physically separated memory]*] operation and that is unambiguously linked with the current content of the transaction with the associated physically separated memory.

FDP_IRA.1.3 The state of the Irreversibility Anchor implemented by the TSF shall be maintained during [selection: *operation, power off, power saving, any operation mode*].

7.1.4 IT Security Requirements

Application Note 42. All SFR comprise an iteration identifier to support the integration in the Protection Profile. If one of the SFRs need to be iterated a digit can added to the current iteration identifier.

7.1.4.1 SFRs for the TOE

The TOE shall meet the requirement "Data Authentication with Identity of Guarantor (FDP_DAU.2)", as specified below.

FDP_DAU.2/PM	**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 external memory[16].
FDP_DAU.2.2/PM	The TSF shall provide the 3S[17] 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/PM, FDP_SDC.1/PM and FDP_SDI.2/PM.**

The TOE shall meet the requirement "Timing of identification (FIA_UID.1)", as specified below.

FIA_UID.1/PM	**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[18] 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 external memory need to be identified before any further action.**

The TOE shall meet the requirement "Replay detection (FPT_RPL.1)", as specified below.

FPT_RPL.1/PM	**Replay detection**
Hierarchical to:	No other components
Dependencies:	No dependencies

16 [assignment: list of objects or information types]
17 [assignment: list of subjects]
18 [assignment: list of TSF-mediated actions]

FPT_RPL.1.1/PM	The TSF shall detect replay for the following entities: commands issued by the 3S to the external memory for the read, write and erase operations[19]
FPT_RPL.1.2/PM	The TSF shall perform [assignment: *list of specific actions*] when a replay is detected.

The TOE shall meet the requirement "Protection against an unauthorised rollback of content (FDP_URC.1)", as specified below.

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 external memory[20] 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 [selection: *stop TOE operation, [assignment: other actions]]*

The TOE shall meet the requirement "Irreversibility Anchor for external memory (FDP_IRA.1)", as specified below.

FDP_IRA.1/PM	**Irreversibility Anchor for external memory**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_IRA.1.1/PM	The TSF shall verify the freshness of data for each read operation from the passive external memory[21].
FDP_IRA.1.2/PM	The Irreversibility Anchor shall maintain a distinct transaction reference for each write, erase[22] operation and that is unambiguously linked with the current content of the transaction with the associated physically separated memory.
FDP_IRA.1.3/PM	The state of the Irreversibility Anchor implemented by the TSF shall be maintained during any operation mode[23].
Refinement:	**The passive external memory is considered outside the TOE, even though it may be packaged together with the SoC including the 3S.**

The TOE shall meet the requirement "Stored data confidentiality (FDP_SDC.1/PM)" as specified below.

FDP_SDC.1/PM	**Stored data confidentiality**

19 [assignment: list of identified entities].
20 [assignment: physically separated memory]
21 [assignment: physically separated memory]
22 [selection: write, erase, *[assignment: further operation that changes the content of the physically separated memory]*]
23 [selection: operation, power off, power saving, any operation mode]

Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SDC.1.1/PM	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the external memory[24].

The TOE shall meet the requirement "Stored data integrity monitoring and action (FDP_SDI.2/PM)" as specified below.

FDP_SDI.2/PM	**Stored data integrity monitoring and action**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1/PM	The TSF shall monitor user data stored in containers controlled by the TSF for *[assignment: integrity errors]* on all objects, based on the following attributes: *[assignment: user data attributes]*.
FDP_SDI.2.2/PM	Upon detection of a data integrity error, the TSF shall [assignment: action to be taken].

7.1.4.2 Rationale for the SFRs

Table 8 below provides an overview, how the SFRs are combined to meet the security objectives. The justification for each objective is detailed after the table.

Objective	TOE Security Functional and Assurance Requirements
O.Pas-Mem-Content-Prot	FDP_SDC.1/PM for confidentiality protection FDP_SDI.2/PM for integrity protection
O.Pas-Mem-Cmd-Replay-Prot	FPT_RPL.1/PM for Replay detection
O.Pas-Mem-Irreversible-Anchor	FDP_IRA.1/PM for Irreversibility Anchor of external memory content
O.Pas-Mem-Unauth-Rollback-Prot	FDP_URC.1/PM for Protection against an unauthorised rollback of content
O.Pas-Mem-Clone-Replace-Prot	FDP_DAU.2/PM for Data Authentication with Identity of Guarantor FIA_UID.1/PM for Timing of identification

Table 8: Mapping between Objectives and SFRs for passive external memory

The justification related to the security objective "Protection against unauthorised disclosure and undetected modification of external memory content (O.Pas-Mem-Content-Prot)" is as follows:

The SFR FDP_SDC.1/PM ensures protection of confidentiality of the content stored in the external memory, while the SFR FDP_SDI.2/PM ensures protection of the integrity of the content stored in the external memory. The protection is under full control inside the 3S, so the transfer between the 3S and the external memory is also protected. Therefore, these SFRs support the objective.

The justification related to the security objective "Protection against replay of commands between the 3S and the external memory (O.Pas-Mem-Cmd-Replay-Prot)" is as follows:

[24] [assignment: *memory area*]

The SFR FPT_RPL.1/PM requires the TSF to detect replayed transactions (read, write and erase operations) to the external memory. This requirement is considered in the assignment of FPT_RPL.1.1/PM. Therefore, this SFR supports the objective. The action on a detected transaction replay is left to the ST author, because it depends on the application context.

The justification related to the security objective “Protection against content (O.Pas-Mem-Unauth-Rollback-Prot)” is as follows:

The SFR FDP_URC.1/PM requires that the TSF detects the case when the content of the external memory has been replaced by previous versions of them. In this way, this SFR supports the objective.

The justification related to the security objective “External memory content Irreversibility Anchor (O.Pas-Mem-Irreversible-Anchor)” is as follows:

The SFR FDP_IRA.1/PM requires the TOE to implement distinct transaction references for each write and erase operation that is unambiguously linked with the current content of the transaction with the external memory. Thereby, the data freshness can be verified during a read operation, based on the data maintained by the irreversible anchor. If the external memory is non-volatile, the Irreversibility Anchor needs to be maintained in any operation mode. By providing the mechanism required by this SFR, the security objective O.Pas-Mem-Irreversible-Anchor is directly supported.

The justification related to the security objective “Protection against external memory cloning or replacement (O.Pas-Mem-Clone-Replace-Prot)” is as follows:

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.

7.1.4.3 Dependencies of SFRs

Requirement	No dependency	Satisfied Dependencies
FDP_SDC.1/PM	No dependency	
FDP_SDI.2/PM	No dependency	
FPT_RPL.1/PM	No dependency	
FDP_IRA.1/PM	No dependency	
FDP_URC.1/PM	No dependency	
FDP_DAU.2/PM	FIA_UID.1	Satisfied by FIA_UID.1/PM

Table 9: Overview of SFR dependencies for passive external memory

All dependencies are satisfied.

7.2 Package for Secure External Memory

This package describes the extension of the security problem definition and the SFRs, if the 3S uses security functionality implemented in the secure external memory which, therefore, is considered to be part of the TOE together with the interface connecting the secure external memory to the 3S. The secure external memory augments the 3S protection mechanisms with its own protection mechanisms

and it is connected to the TOE using a secure interface. This configuration has the following implications:

- The TOE implements an external interface to access the secure external memory
- The TOE establishes a secure interface between the 3S and the secure external memory using a unique binding key.
- The secure external memory provides additional security functionality to protect code and data stored in the secure external memory.
- The FW, SW and Composite Software stored in the secure external NVM before the SoC is deployed in the field is authenticated before being pre-programmed to the secure external NVM. Therefore, the FW, SW and Composite Software can be executed after loading from the secure external NVM using the integrated security mechanisms.

Application Note 43. The secure external memory can either be evaluated as part of the TOE or the secure external memory can be evaluated independent of the 3S. If the secure external memory is evaluated independent of the 3S, these evaluation results can be used to integrate the secure external memory as part of the TOE during the evaluation of the 3S using the composite evaluation approach defined for the 3S.

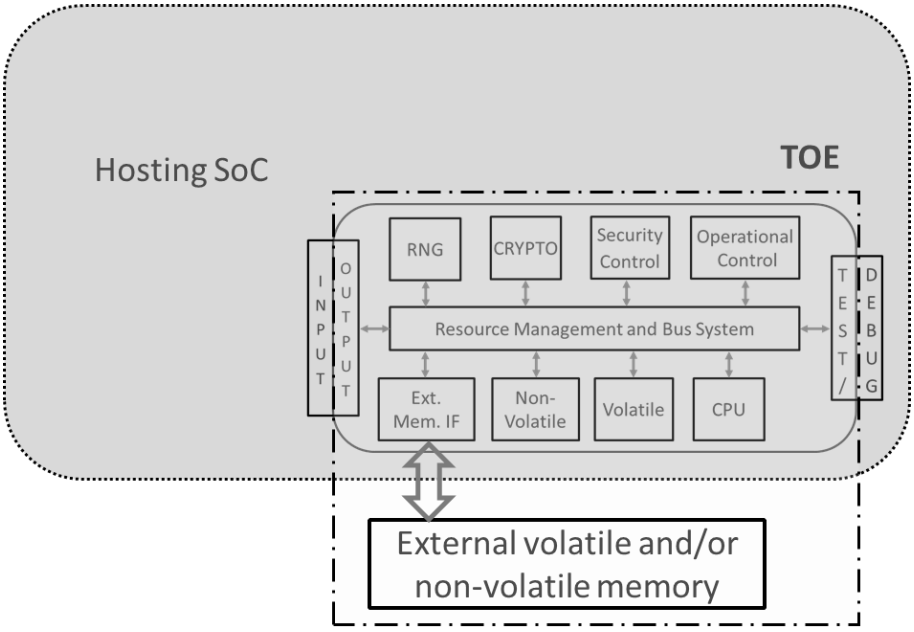


Figure 8: 3S with secure external memory

The secure external memory is part of the TOE, as well as the communication bus connecting the secure external memory to the 3S. Those provide dedicated security functionality to protect the data stored inside the secure external memory. The protection of the secure external memory needs to cover confidentiality, integrity and replay protection for code and data stored in the secure external memory. The protection is modelled with the functional package “Secure External Memory”.

7.2.1 Security Problem Definition

7.2.1.1 Description of Assets

Application Note 44. There are no additional assets defined in this package.

7.2.1.2 Threats

The following figure describes the attacks on the TOE with secure external memory included in TOE. The threats described in this section shall be added in the Security Target together with the threats against the TOE defined for the base configuration (see section 3.2).

The attacks marked in blue are applicable only for the configuration with secure external memory. The threats defined in this Protection Profile shall be averted by the combination of the security functionality implemented by the 3S and security functionality implemented by the secure external memory.

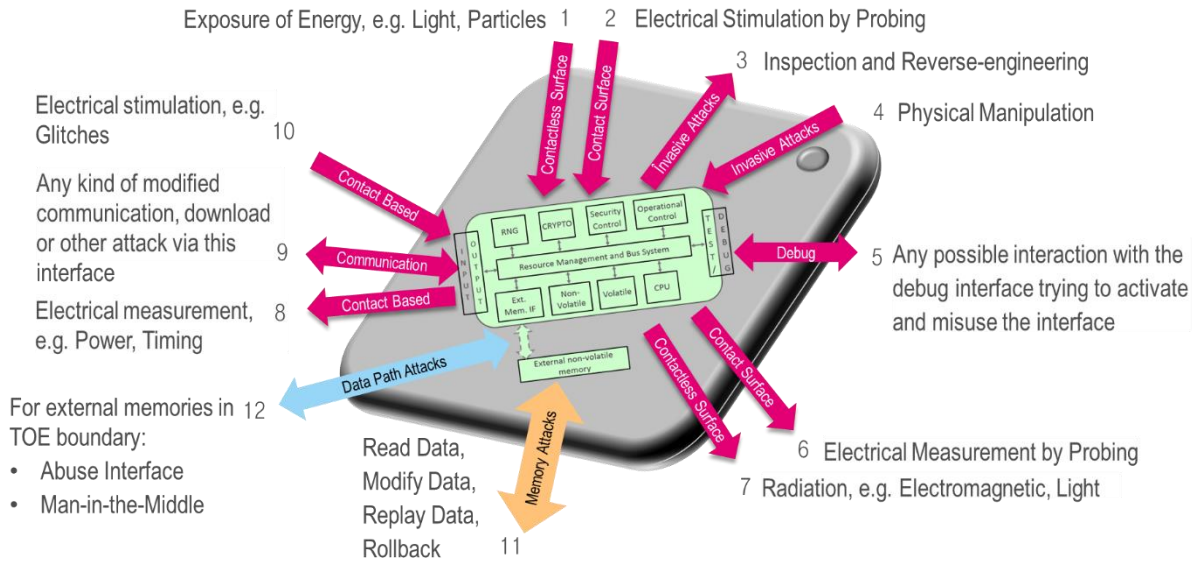


Figure 9: Attacks against secure external memory

In Figure 9, the grey box represents the SoC with the TOE (green box) and its interaction channels. The secure external memory also implements security functionality and is part of the TOE. The orange arrows denote attacks to the content of the external memory, while the blue arrows denote attacks to the interface between 3S and external memory.

Application Note 45. The external memory may be stacked on the SoC or embedded in a separate package. This has no relevant impact on the attacks described in this section.

The TOE shall avert the threat “Cloning the TOE with a Copy of the external memory (T.Sec-Mem-Clone-Replace)” as specified below.

T.Sec-Mem-Clone-Replace	Cloning or replacement of secure external memory An attacker may attempt to clone the full content of the external memory or the 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.

This threat refers to the case where the full content of the external memory is cloned to a different device. It can also cover the replacement of the physical external memory used by the 3S with a different memory unit. The second case might not be viable on some architectures when the physical design or assembly procedures impede it.

The effect of this threat is in replacing the data and/or image of a TOE with a different one and to obtain a valid but unauthorised instance of the TOE.

This threat involves using two different TOE units or instances. One TOE unit is used as a source for the external memory content. This content is used to replace the genuine content of the external memory of the second TOE unit.

Another possible scenario for this threat can be contemplated: the external memory is replaced with an empty or virgin unit, removing the user and TSF data used by the TOE, and possibly forcing the TSF to generate new user and TSF data, potentially affecting the TSF behaviour.

The TOE shall avert the threat "Abuse of external memory content (T.Sec-Mem-Content-Abuse)" as specified below.

T.Sec-Mem-Content-Abuse	Abuse of 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.

An attacker may obtain unauthorised access to the external memory and attempt to read, disclose, modify or replace the content of the external memory. This threat addresses also the authenticity of the data stored in the external memory.

Note that the access to the external memory or the transfer of data between the TOE and the external memory may also support an attack.

The TOE shall avert the threat "Replay of commands between the 3S and the external memory (T.Sec-Mem-Cmd-Replay)" as specified below.

T.Sec-Mem-Cmd-Replay	Replay of commands between the 3S and the external memory An attacker may attempt to replay the write and erase commands or responses to the read commands between the 3S and the external memory, to affect the freshness of the content read from or written to the external memory.

The read, write and erase commands issued by the 3S to exercise the memory functionality of the external memory, and their payloads, can be intercepted by an attacker (e.g., eavesdrop the commands on the link between the 3S and the external memory). Such an attacker may use copies of these commands to try to misuse the TOE or compromise data. The command replay attack can take the following forms:

- The attacker reacts to a read command and replies with a previously recorded answer (e.g., to a previous read request). In this way, the 3S gets an old version of such content.
- The attacker issues a previous write command, trying to overwrite the external memory with the previous content, and leading to the 3S obtaining old versions of such content in later read operations.
- The attacker issues a previous erase command, trying to overwrite status information or other data that may lead to misuse of the TOE.

The TOE shall avert the threat "Unauthorised rollback of content in the secure external memory (T.Sec-Mem-Unauth-Rollback)" as specified below.

T.Sec-Mem-Unauth-Rollback	Unauthorised rollback of content in the secure 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 was updated by the TOE.

This threat takes advantage of the fact that the external memory is not integrated into the 3S. Therefore, physical protections for preventing the replacement of stored content may not cover the external memory. This situation may enable an attacker to read and write the content of the external memory. Even if the confidentiality and integrity of the external memory content is protected, the replacement with an old copy may also be valid, because it is retrieved from the external memory.

If the TOE image is stored in the external memory, this threat may lead to an unauthorised rollback of the TOE image to an older version. Even when the TOE stores data and not code in the external memory, this data rollback might affect the behaviour of the TSF.

The replacement of content stored in the external memory with previous versions of it may refer to the full content of the external memory or partial content of it, depending on the organization and protection of the data stored in the external memory.

The TOE shall avert the threat "Abuse of interface between 3S and secure external memory (T.Sec-Mem-Abuse-Interface)" as specified below.

T.Sec-Mem-Abuse-Interface: Abuse of interface between 3S and secure external memory

An attacker may abuse the link or the interface between the 3S and the secure external memory to (i) disclose the user data and/or TSF data in transit, (ii) manipulate the user data and/or TSF data in transit, (iii) block commands or issue commands for modification of the secure external memory content.

This threat covers attacks on read, write and erase operations happening between the 3S and the secure external memory. The operations can be blocked or intercepted by an attacker eavesdropping to the interconnection bus (e.g., by a man-in-the-middle attack), to disclose the user data and/or TSF data being written to or read from the secure external memory before security services are executed or finalised by the secure external memory.

7.2.1.3 Organisational Security Policies

Application Note 46. This package does not define any additional organisational security policy.

7.2.1.4 Assumption

The following assumption shall be added in the Security Target only, if the 3S is connected to secure external memory

The SoC Integrator shall fulfil the assumption "Usage and binding of Secure External memory (A.Ext-SecMem)" as specified below.

A.Ext-SecMem: Usage and binding of Secure External memory

It is assumed that the SoC Integrator integrates a secure external memory. The secure external memory shall be unambiguously linked to a 3S using a unique binding key during the integration. This binding key enables the secure connection between the secure external memory and the 3S to be protected against cloning, replacement and rollback.

The connection between the 3S and the secure external memory requires a secure interface. This secure interface is established, based on the unique binding key configured during the initialisation of the two components.

7.2.2 Security Objectives

7.2.2.1 Security Objectives for the TOE

The TOE shall provide "Protection of external Content (O.Sec-Mem-Content-Prot)" as specified below.

O.Sec-Mem-Content-Prot: Protection against disclosure and undetected modification of external memory content.

The content in the external memory shall be protected against disclosure and undetected modification, because an attacker can directly access the external memory.

This security objective requires protection of the content stored in external memory by the TOE. The protection prevents disclosure and identifies modifications of stored code and data that is not performed by the TOE.

The TOE shall provide "Protection against replay of commands to store or modify data in the secure external memory to the 3S (O.Sec-Mem-Cmd-Replay-Prot)" as specified below.

O.Sec-Mem-Cmd-Replay-Prot: Protection against replay of commands to store or modify data in the secure external memory to the 3S.

The TOE shall protect against replay of content during write, read and erase operations to the external memory by the 3S.

This security objective requires protection against replay of read, write and erase operations. This covers simple replay of previously recorded commands or memory content but also the replay of modified commands or memory content. The TOE shall be able to detect such attacks violating the TOE.

The TOE shall provide "Protection against an unauthorised rollback of secure external memory content (O.Sec-Mem-Unauth-Rollback-Prot)" as specified below.

O.Sec-Mem-Unauth-Rollback-Prot: Protection against an unauthorised rollback of secure external memory content.

The TOE shall protect against replacement of the external memory content with a previous version, even if it was valid in the past.

The security objective requires protection against the simulation of outdated content. Replacement of memory content with a previous version of the same memory content or the manipulations of write operations violate the freshness of the external memory content and shall be detected by the TOE.

The TOE shall provide "Secure external memory content Irreversibility Anchor (O.Sec-Mem-Irreversible-Anchor)" as specified below.

O.Sec-Mem-Irreversible-Anchor Secure external memory content Irreversibility Anchor

The TOE shall implement a reference 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.

The security objective requires the verification of freshness for data read from the external memory. Therefore, the 3S shall maintain a reference that represents the current content of the external memory. This reference needs to be updated with each authorised read and write operation to detect a violation of the data freshness. It should be maintained in any TOE operational state, including the standby and sleep states. In the case of non-volatile memory, the Irreversibility Anchor needs to be persistently saved between two boots.

The TOE shall provide "Protection against secure external memory cloning or replacement (O.Sec-Mem-Clone-Replace-Prot)" as specified below.

O.Sec-Mem-Clone-Replace-Prot: Protection against secure external memory cloning or replacement.

The TOE shall protect against cloning or replacement of content with the content 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.

The security objective requires protection against replacement of its external memory content with the content of another instance of the TOE. The external memory content shall only be valid for the 3S that is initially linked to this external memory. The replacement of the external memory or the transfer of the content from a memory unit that is linked to another instance of the TOE shall be detected.

The TOE shall provide "Protection against abuse of the interface between 3S and secure external memory (O.Sec-Mem-Interface-Prot)", as specified below.

O.Sec-Mem-Interface-Prot: Protection against abuse of the interface between 3S and secure external memory

The TOE shall protect the data in transit between the 3S and the external memory against disclosure. The TOE shall also detect manipulation of the data in transit through the interconnection bus and manipulation through issuing commands to the external memory.

7.2.2.2 Security Objectives for the TOE Environment

OE.Ext-SecMem: Binding between 3S and Secure External memory

The binding between the 3S and the Secure External memory is set up in a trustworthy production environment. This comprises the initial key exchange and the related initialisation.

7.2.2.3 Security Objectives Rationale

	O.Sec-Mem-Content-Prot	O.Sec-Mem-Cmd-Replay-Prot	O.Sec-Mem-Irreversible-Anchor	O.Sec-Mem-Unauth-Rollback-Prot	O.Sec-Mem-Clone-Replace-Prot	O.Sec-Mem-Interface-Prot	OE.Ext-SecMem
T.Sec-Mem-Content-Abuse	X						
T.Sec-Mem-Cmd-Replay		X	X				
T.Sec-Mem-Unauth-Rollback			X	X			
T.Sec-Mem-Clone-Replace					X		
T.Sec-Mem-Abuse-Interface						X	
A.Ext-SecMem							X

Table 10: Mapping between objectives and threats

In the following, the justification of the coverage of the threats and organisational security policies by the security objectives is given.

T.Sec-Mem-Content-Abuse is countered by O.Sec-Mem-Content-Prot, which requires the TOE to prevent disclosure and undetected modification of the content stored in external memory.

T.Sec-Mem-Cmd-Replay is countered by O.Sec-Mem-Cmd-Replay-Prot and O.Sec-Mem-Irreversible-Anchor as follows:

- O.Sec-Mem-Cmd-Replay-Prot requires protection against replay of commands exported from the 3S in the external memory mitigating T.Sec-Mem-Cmd-Replay.
- O.Sec-Mem-Irreversible-Anchor requires the implementation of a reference representing the current content of the external memory. The reference 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.Sec-Mem-Unauth-Rollback is countered by O.Sec-Mem-Unauth-Rollback-Prot and O.Sec-Mem-Irreversible-Anchor as follows:

- O.Sec-Mem-Unauth-Rollback-Prot requires that the TOE protects against replacement of external memory content with older content of the same memory, where the data freshness property is not met, thereby mitigating this threat.
- O.Sec-Mem-Irreversible-Anchor requires that the TOE implements a reference representing the current content of the external memory. The reference 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.Sec-Mem-Clone-Replace is countered by O.Sec-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 a unit of a different instance of the TOE.

T.Sec-Mem-Abuse-Interface is countered by O.Sec-Mem-Interface-Prot, which requires the TOE to prevent disclosure and detect modification of the data in transit between the 3S and the external memory.

The justification related to the organisational security policy "Usage and binding of Secure External memory (A.Ext-SecMem)" is as follows:

OE.Ext-SecMem requires the use of secure external memory and the binding between the 3S and the secure external memory. Therefore, initial key exchange and the initialisation of the connection shall be performed in a trustworthy environment. The assumption A.Ext-SecMem addresses this objective, because the usage of secure external memory and a secure binding is assumed.

7.2.3 Extended Component Definition

Application Note 47. The same Extended SFRs need to be added in the definition of this package. The extended component definition is only reference here to support consistency between the two packages.

7.2.3.1 Definition of the Family FDP_URC

Application Note 48. Add the definition of the Extended SFR in section 7.1.3.1.

7.2.3.2 Definition of the Family FDP_IRA

Application Note 49. Add the definition of the Extended SFR in section 7.1.3.2.

7.2.4 IT Security Requirements

7.2.4.1 SFRs for the TOE

The TOE shall meet the requirement "Data Authentication with Identity of Guarantor (FDP_DAU.2)", as specified below.

FDP_DAU.2/SM	**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/SM	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 external memory[25].
FDP_DAU.2.2/SM	The TSF shall provide the 3S[26] 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 user generating the evidence is the dedicated 3S instance for any user data stored in the external secure memory.**

The TOE shall meet the requirement "Timing of identification (FIA_UID.1)", as specified below.

FIA_UID.1/SM	**Timing of identification**
Hierarchical to:	No other components.
Dependencies:	No dependencies.

[25] [assignment: list of objects or information types]
[26] [assignment: list of subjects]

FIA_UID.1.1/SM	The TSF shall allow the secure start-up or wake-up without access to user data[27] on behalf of the user to be performed before the user is identified.
FIA_UID.1.2/SM	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
Refinement:	**Instead, the identification of the user, the identification of the unambiguously-assigned external secure memory is required before further actions are performed. Based on the unambiguous assignment only one instance of the external secure memory can be identified as valid.**

The TOE shall meet the requirement "Replay detection (FPT_RPL.1)", as specified below.

FPT_RPL.1/SM	**Replay detection**
Hierarchical to:	No other components
Dependencies:	No dependencies
FPT_RPL.1.1/SM	The TSF shall detect replay for the following entities: commands issued by the 3S to the external memory for the read, write and erase operations[28].
FPT_RPL.1.2/SM	The TSF shall perform [assignment: list of specific actions] when a replay is detected.

The TOE shall meet the requirement "Protection against an unauthorised rollback of memory content (FDP_URC.1)", as specified below.

FDP_URC.1/SM	**Protection against an unauthorised rollback of memory content**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_URC.1.1/SM	The TOE shall detect an unauthorised replacement of the contents stored in external **memory**[29] before the contents are 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/SM	Upon detection of unauthorised rollback of the content stored in a physically separated memory, the TOE shall *[selection: stop TOE operation, [assignment: other actions]]*

The TOE shall meet the requirement "Irreversibility Anchor for external memory (FDP_IRA.1)", as specified below.

FDP_IRA.1/SM	**Irreversibility Anchor for external memory**
Hierarchical to:	No other components.
Dependencies:	No dependencies.

[27] [assignment: list of TSF-mediated actions]
[28] [assignment: list of identified entities].
[29] [assignment: physically separated storage]

FDP_IRA.1.1/SM	The TSF shall verify the freshness of data for each read operation from the external memory[30].
FDP_IRA.1.2/SM	The Irreversibility Anchor shall maintain a distinct transaction references for each write, erase[31] operation and that is unambiguously linked with the current content of the transaction with the associated physically separated memory.
FDP_IRA.1.3/SM	The state of the Irreversibility Anchor implemented by the TSF shall be maintained during any operation mode[32].

The TOE shall meet the requirement "Stored data confidentiality (FDP_SDC.1/SM)" as specified below.

FDP_SDC.1/SM	**Stored data confidentiality**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SDC.1.1/SM	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the secure external memory[33].

The TOE shall meet the requirement "Stored data integrity monitoring and action (FDP_SDI.2/SM)" as specified below.

FDP_SDI.2/SM	**Stored data integrity monitoring and action**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
Dependencies:	No dependencies.
FDP_SDI.2.1/SM	The TSF shall monitor user data stored in containers controlled by the TSF for *[assignment: integrity errors]* on all objects, based on the following attributes: *[assignment: user data attributes]*.
FDP_SDI.2.2/SM	Upon detection of a data integrity error, the TSF shall [assignment: action to be taken].

Application Note 50. FPT_EMS.1 defines the protection of data in transit between 3S and secure external memory as well as the processing of data inside the secure external memory. This comprises confidentiality and integrity of the TSF data as well as confidentiality and integrity of the user data.

FPT_EMS.1/SM	**Emanation of TSF and user data**
Hierarchical to:	No other components.
Dependencies:	No dependencies.

30 [assignment: physically separated storage]

31 [selection: write, erase, *[assignment: further operation that changes the content of the physically separated memory]*]

32 [selection: operation, power off, power saving, any operation mode]

33 [assignment: *memory area*]

FPT_EMS.1.1/SM The TSF shall ensure that the TOE does not emit emissions over its attack surface to such an extent that these emissions enable access to TSF data and user data, as specified in the following table:

ID	Emanation	Attack Surface	TSF data	User Data
1	[assignment: *list of types of emissions*]	[assignment: *list of types of attack surface*]	[assignment: *list of types of TSF data*]	[assignment: *list of types of user data*]

Table 11: Definition of FPT_EMS

Application Note 51. If FPT_EMS.1.1/SM instantiation is the same as in the base PP, in the security target it can be included only once.

7.2.4.2 Rationale for the SFRs

Table 12 below provides an overview, how the SFRs are combined to meet the security objectives. The justification for each objective is detailed after the table.

Objective	TOE Security Functional and Assurance Requirements
O.Sec-Mem-Content-Prot	FDP_SDC.1/SM for confidentiality protection FDP_SDI.2/SM for integrity protection
O.Sec-Mem-Cmd-Replay-Prot	FPT_RPL.1/SM for Replay detection
O.Sec-Mem-Irreversible-Anchor	FDP_IRA.1/SM for Irreversibility Anchor of external memory content
O.Sec-Mem-Unauth-Rollback-Prot	FDP_URC.1/SM for Protection against an unauthorised rollback of memory content
O.Sec-Mem-Clone-Replace-Prot	FDP_DAU.2/SM for Data Authentication with Identity of Guarantor FIA_UID.1/SM for Timing of identification
O.Sec-Mem-Interface-Prot	FPT_EMS.1/SM Emanation of TSF and user data

Table 12: Mapping between Objectives and SFRs for secure external memory

The SFR FDP_SDC.1/SM and FDP_SDI.2/SM defined in this protection provide support the objective O.Sec-Mem-Content-Prot.

The justification related to the security objective “Protection against unauthorised disclosure and undetected modification of external memory content (O.Sec-Mem-Content-Prot)” is as follows:

The SFR FDP_SDC.1/SM ensures protection of confidentiality of the content stored in the external memory, while the SFR FDP_SDI.2 ensures protection of the integrity of the content stored in the external memory. The protection is under full control inside the 3S, so the transfer between the 3S and the external memory is also protected. Therefore, these SFRs support the objective.

The justification related to the security objective “Protection against replay of commands between the 3S and the external memory (O.Sec-Mem-Cmd-Replay-Prot)” is as follows:

The SFR FPT_RPL.1/SM requires the TSF to detect replayed transactions (read, write and erase operations) to the external memory. This requirement is considered in the assignment of FPT_RPL.1.1/SM. Therefore, this SFR supports the objective. The action on a detected transaction replay is left to the ST author, because it depends on the application context.

The justification related to the security objective “Protection against content (O.Sec-Mem-Unauth-Rollback-Prot)” is as follows:

The SFR FDP_URC.1/SM requires that the TSF detects the case when the content of the external memory has been replaced by previous versions of them. In this way, this SFR supports the objective.

The justification related to the security objective “External memory content Irreversibility Anchor (O.Sec-Mem-Irreversible-Anchor)” is as follows:

The SFR FDP_IRA.1/SM requires the TOE to implement distinct transaction references for each write and erase operation that is unambiguously linked with the current content of the transaction with the external memory. Thereby, the data freshness can be verified during a read operation, based on the data maintained by the irreversible anchor. The Irreversibility Anchor needs to be maintained in any operation mode. By providing the mechanism required by this SFR, the security objective O.Sec-Mem-Irreversible-Anchor is directly supported.

The justification related to the security objective “Protection against external memory cloning or replacement (O.Sec-Mem-Clone-Replace-Prot)” is as follows:

The SFR FDP_DAU.2/SM requires the TOE to be able to generate evidence that guarantees the validity of data objects and containers stored in the external memory. With the refinement that the dedicated 3S instance is the user in case of user data, the cloning or replacement of the external memory is detected. The SFR FIA_UID.1/SM requires the definition of actions that can be performed without user identification. Here the external memory needs to be identified instead of a user. This is described in a refinement for this SFR. The external memory needs to be identified before any user data is accessed. By providing the mechanism required by these two SFRs, the security objective O.Sec-Mem-Clone-Replace-Prot is directly supported.

The justification related to the security objective “Protection against abuse of the interface between 3S and secure external memory (O.Sec-Mem-Interface-Prot)” is as follows:

FPT_EMS.1/SM requires the TOE to protect TSF data and user data when transferred between the 3S and the secure external memory and during processing in the secure external memory. Therefore, this SFR addresses the security objective.

7.2.4.3 Dependencies of the SFRs

Requirement	No dependency	Satisfied Dependencies
FDP_SDC.1/SM	No dependency	
FDP_SDI.2/SM	No dependency	
FPT_RPL.1/SM	No dependency	
FDP_IRA.1/SM	No dependency	
FDP_URC.1/SM	No dependency	
FDP_DAU.2/SM	FIA_UID.1	Satisfied by FIA_UID.1/SM
FPT_EMS.1/SM	No dependency	

Table 13: Overview of Dependencies of the SFRs for secure external memory

7.3 Package for Loader Functionality

7.3.1 Security Problem Definition

7.3.1.1 Description of Assets

Application Note 52. There are no additional assets defined in this package.

7.3.1.2 Threat

Application Note 53. No new threat is defined in this package while all threats of the base Protection Profile are applicable to the loader package.

7.3.1.3 Organisational Security Policies

The Loader Package defines a secure loading process. The ST shall include this package if the Loader can be used after delivery of the TOE including the operational phase.

This package supports access control on usage of the Loader, mutual authentication of the TOE and the authorised user as end-points of a trusted channel and protection of integrity and confidentiality of the data downloaded to the TOE.

P.Access-Ctlr-Loader	Loader Functionality with User Authorisation Authorised user controls the usage of the Loader functionality in order to protect user data stored and loaded to the TOE from disclosure and manipulation.

7.3.1.4 Assumption

Application Note 54. This package does not define an additional assumption.

7.3.2 Security Objectives

7.3.2.1 Security Objectives for the TOE

The TOE shall provide “Access control and authenticity for the Loader (O.Ctrl-Auth-Loader)” as specified below.

O.Ctrl-Auth-Loader	Access control and authenticity for the Loader The TSF provides trusted communication channel with authorised user, supports confidentiality protection and authentication of the user data to be loaded and access control for usage of the Loader functionality.

7.3.2.2 Security Objectives for the Environment

The operational environment of the TOE shall provide “Secure communication and usage of the Loader (OE.Loader-Usage)” as specified below.

OE.Loader-Usage	Secure communication and usage of the Loader The authorised user shall support a trusted communication channel with the TOE which protects confidentiality and proofs authenticity of data to be loaded and fulfilling the access conditions required by the Loader.

7.3.2.3 Security Objectives Rationale

Table 14: Mapping overview between objectives and threats respectively policies

The organisational security policy "Controlled usage to Loader Functionality (P.Access-Ctlr-Loader) is directly implemented by the security objective for the TOE "Access control and authenticity for the Loader (O.Ctrl-Auth-Loader)" and the security objective for the TOE environment "Secure communication and usage of the Loader (OE.Loader-Usage)".

7.3.3 Extended Component Definition

Application Note 55. This package does not define additional extended components.

7.3.4 IT Security Requirements

7.3.4.1 SFRs for the TOE

The TOE shall meet the requirement "Inter-TSF trusted channel (FTP_ITC.1)" is specified as follows.

FTP_ITC.1/Load	**Inter-TSF trusted channel**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FTP_ITC.1.1/Load	The TSF shall provide a communication channel between itself and [assignment: *users authorised for using the Loader*] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2/Load	The TSF shall permit another trusted IT product to initiate communication via the trusted channel.
FTP_ITC.1.3/Load	The TSF shall initiate communication via the trusted channel for deploying Loader [assignment: *rules*].

The TOE shall meet the requirement "Basic data exchange confidentiality (FDP_UCT.1)" is specified as follows.

FDP_UCT.1/Load	**Basic data exchange confidentiality**

Hierarchical to:	No other components.
Dependencies:	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]
	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_UCT.1.1/Load	The TSF shall enforce the Loader SFP to receive user data in a manner protected from unauthorised disclosure.

The TOE Functional Requirement "Data exchange integrity (FDP_UIT.1)" is specified as follows.

FDP_UIT.1/Load	**Data exchange integrity**
Hierarchical to:	No other components.
Dependencies:	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]
	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_UIT.1.1/Load	The TSF shall enforce the Loader SFP to receive user data in a manner protected from modification, deletion, insertion errors.
FDP_UIT.1.2/Load	The TSF shall be able to determine on receipt of user data, whether modification, deletion, insertion has occurred.

The TOE shall meet the requirement "Subset access control - Loader (FDP_ACC.1/Load)" is specified as follows.

FDP_ACC.1/Load	**Subset access control - Loader**
Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control.
FDP_ACC.1.1/Load	The TSF shall enforce the Loader SFP on the following: (1) the subjects [assignment: *authorised roles for using Loader*], (2) the objects user data in [assignment: *memory areas*], (3) the operation deployment of Loader.

Application Note 56. The TOE enforces the Loader SFP by FTP_ITC.1, FDP_UCT.1 and FDP_UIT.1 and FDP_ACF.1 to describe additional access control rules.

The TOE shall meet the requirement "Security attribute based access control - Loader (FDP_ACF.1/Load)" is specified as follows.

FDP_ACF.1/Load	**Security attribute based access control - Loader**
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.3 Static attribute initialisation FDP_ACF.1.1/Load
FDP_ACF.1.1/Load	The TSF shall enforce the Loader SFP[34] to objects, based on the following:

[34] [assignment: access control SFP]

(1) the subjects [assignment: *authorised roles for using Loader*] with security attributes [assignment: *SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]

(2) the objects [assignment: *user data in memory areas*] with security attributes [assignment: *SFP-relevant security attributes, or named groups of SFP-relevant security attributes*][35].

FDP_ACF.1.2/Load FDP_ACF.1.2 The TSF shall enforce the following rules to determine whether an operation among controlled subjects and controlled objects is allowed: [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*].

FDP_ACF.1.3/Load FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects, based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].

FDP_ACF.1.4/Load The TSF shall explicitly deny access of subjects to objects, based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

Application Note 57. The ST writer shall perform the open operations in the component of FDP_ACF.1/Load, to describe additional access control rules. The open assignment of security attributes may be empty.

The ST writer may define the dependent SFR FMT_MSA.3, if management of the relevant security attributes is implemented for the Loader SFP.

7.3.4.2 Rationale for the SFRs

Objective	TOE Security Functional and Assurance Requirements	
O.Ctrl-Auth-Loader	FTP_ITC.1	Inter-TSF trusted channel
	FDP_UCT.1	Basic data exchange confidentiality
	FDP_UIT.1	Data exchange integrity
	FDP_ACC.1/Load	Subset access control – Loader
	FDP_ACF.1/Load	Security attribute based access control - Loader

Table 15: Mapping between Objectives and SFRs for the Loader

The security objective Access control and authenticity for the Loader (O.Ctrl-Auth-Loader) is covered by the SFR as follows:

The SFR FDP_ACF.1/Load and FDP_ACC.1.1/Load require the TSF to implement access control for the Loader functionality.

The SFR FTP_ITC.1, FDP_UCT.1 and FDP_UIT.1 require the TSF to establish a trusted channel with assured identification of its end points, encryption and protection of the channel data from modification or disclosure.

[35] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

7.3.4.3 Dependencies of the SFRs

Requirement	No dependency	Satisfied Dependencies
FTP_ITC.1/Load	No dependency	
FDP_UCT.1/Load	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FTP_ITC.1/Load and FDP_ACC.1/Load
FDP_UIT.1/Load	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	FTP_ITC.1/Load and FDP_ACC.1/Load
FDP_ACC.1/Load	FDP_ACF.1	FDP_ACF.1/Load
FDP_ACF.1/Load	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Load FMT_MSA.3/Load
FMT_MSA.3/Load	FMT_MSA.1 FMT_SMR.1	The dependencies are not satisfied, see the rationale below the table

Table 16: Overview of SFR dependencies for the Loader package

The SFRs FMT_MSA.1 and FMT_SMR.1 are not defined, because the security attributes shall not be changed. Each software image loaded in the TOE shall be checked and verified in the same way. Therefore, no functionality and no role are required to manage the security attributes.

7.4 Crypto Package

This section defines a general optional package for cryptographic services that may be provided by a TOE.

7.4.1 Security Problem Definition

7.4.1.1 Description of Assets

The assets are covered by the asset description in the base PP.

7.4.1.2 Threats

No new threats are included in this package while all threats of the base Protection Profile are applicable to these cryptographic services.

7.4.1.3 Organisational Security Policies

The cryptographic security services described in this package implement the organizational security policy comprising a list with the implemented cryptographic services. The use of this services by the Composite Software is optional.

The TOE shall implement the policy “Cryptographic service of the TOE (P.Crypto-Service)” as specified below.

P.Crypto-Service — Cryptographic service of the TOE

The TOE provides secure platform based cryptographic services that can be used by the Composite Software.

Application Note 58. The organizational security policy P.Crypto-Service shall be implemented by separate security objectives for each cryptographic service. Each security objective can be directly implemented by specific SFR of the class “Cryptographic Support”. This may be pure hardware implementation of the cryptographic algorithm or more complex combination of hardware and FW/SW. The cryptographic services may be provided as library functions that need to be compiled together with the Composite Software or as API that is used by the Composite Software.

7.4.1.4 Assumption

This package does not define an additional assumption.

7.4.2 Security Objectives

7.4.2.1 Security Objectives for the TOE

The TOE shall provide the “Cryptographic service (O.Crypto_Service)” as specified below.

O.Crypto_Service	Cryptographic Algorithm The TOE provides the cryptographic algorithm for the selected cryptographic operations and the selected modes of operation for the following “purpose”.

The security objectives listed under “Cryptographic service (O.Crypto_Service)” enforces the organizational security policy P.Crypto-Service.

The term “purpose” shall be replaced by the intended usage of the cryptographic algorithm.

7.4.2.2 Security Objectives for the TOE Environment

This package does not include additional Security Objectives for the TOE Environment.

7.4.2.3 Security Objectives Rationale

	O.Crypto_Services
P.Crypto-Service	X

Table 17: Mapping between OSP and objectives

The organisational security policy “Cryptographic services of the TOE (P.Crypto-Service) is directly implemented by the security objective(s) for the TOE “Cryptographic Algorithm (O.Crypto_Service)”.

7.4.3 Extended Component Definition

This package does not define additional extended components.

7.4.4 IT Security Requirements

7.4.4.1 SFRs for the TOE

The TOE shall meet the requirement "Cryptographic operation of the selected algorithm (FCS_COP.1/iteration" as specified below.

FCS_COP.1/iteration	**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/iteration	The TSF shall perform [assignment: *list of cryptographic operations*] in accordance with a specified cryptographic algorithm [assignment: *cryptographic algorithm*] and cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*].

Application Note 59. The term "iteration" in the FCS_COP1 definition above shall be replaced by an identifier for the algorithm defined by the SFR. The iteration allows the definition of several cryptographic algorithms associated with the security objectives. If only one cryptographic algorithm is added in the Security Target the iteration identifier is not required.

Application Note 60. The cryptographic operations defined in [7] include cryptographic algorithms according to standards accepted by various certification bodies. The use of such crypto algorithms supports the re-use of evaluation results for higher assurance levels.

The TOE shall meet the requirement "Cryptographic key destruction (FCS_CKM.4/iteration" as specified below.

FCS_CKM.4/iteration	**Cryptographic key destruction**
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	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

Application Note 61. The ST author shall provide iterations of FCS_CKM.4 for any of the selected key destruction method. The term "iteration" shall be replaced by an appropriate term for the identification of the specified destruction method. If only one algorithm is added in the Security Target the iteration identifier is not required. Depending on the implemented key storage and the define key destruction

method, the definition of one SFR for FCS_CKM.4 can meet the dependency for various cryptographic algorithms defined with FCS_COP1.

7.4.4.2 Rationale for the SFRs

The FCS_COP.1 and FCS_CKM.4 meet the security objective "Cryptographic service (O.Crypto_Services)".

7.4.4.3 Dependencies of the SFRs

Requirement	No dependency	Satisfied Dependencies
FCS_COP.1/iteration	FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1 FCS_CKM.4	FCS_CKM.4/iteration
FCS_CKM.4/iteration	FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1	

Table 18: Overview of SFR dependencies for the Loader package

The dependency of FCS_COP.1 on FCS_CKM.4 is fulfilled within the package

FCS_COP.1 and FCS_CKM.4 have a dependency to [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]. This PP leave the decision to the ST author not preferring one of the alternative methods as source for the keys. Therefore, the ST author shall include the respective SFR component.

7.5 Composite Software Isolation Package

This package defines additional security functionality to enable the separation between different software packages. These software packages may be delivered by different composite software developers.

7.5.1 Security Problem Definition

7.5.1.1 Description of Assets

Application Note 62. The assets are covered by the asset description in the base PP

7.5.1.2 Threats

Application Note 63. This package does not define an additional threat beyond the threats of the base PP

7.5.1.3 Organisational Security Policy

P.Access-Ctrl-to-TSF — TSF access control against unauthorised access to TSF from any user

The TSF shall perform access control to TSF resources to ensure that only authorised and known subjects running on TSF can access the associated code and data.

P.Access-Ctrl-to-Composite-SW — TSF access control against unauthorised access to Composite Software

TSF shall perform access control to Composite Software to avoid any unauthorised access to Composite Software (code and data) by unauthorised or unknown TSF processes or subjects running on TSF.

7.5.1.4 Assumption

Application Note 64. This package does not define an additional assumption.

7.5.2 Security Objectives

7.5.2.1 Security Objectives for the TOE

O.TSF-Access	Access and Operation control on TSF data The TOE permits Composite Software to only have access to TSF data, security services and hardware resources that are intended to be accessed by the Composite Software. The TOE protects TSF data that shall not be accessible to Composite Software. In addition, a privileged mode shall define access to hardware resources for processes running in unprivileged operation mode.
O.Mem-Access	Access control on memory and hardware resources The TOE shall control access of processes (CPU, DMA, etc) to memory areas to separate code and data owned by different entities. The TOE shall provide the capability to limit access to code and data for processes running in unprivileged operation mode. Further on, the TOE shall provide a privileged operation mode with the capabilities to configure memory partitions and associated access properties for the unprivileged operation mode. The access control shall separate Composite Software applications[36] running on behalf of different entities. If such Composite Software applications are simultaneously processed, the code running on behalf of one user shall not be impacted by any code running on behalf of another user. In addition, the sequential use of security services and/or hardware resources shall not leak any data between Composite Software applications running on behalf of different entities, and shall prevent the re-use of data processed by different entities.

7.5.2.2 Security Objectives for the TOE Environment

Application Note 65. This package does not include additional Security Objectives for the TOE Environment.

[36] Composite Software applications means software packages or software components that may be provided by different developers.

7.5.2.3 Security Objectives Rationale

	O.TSF-Access	O.Mem-Access
P.Access-Ctrl-to-TSF	X	
P.Access-Ctrl-to-Composite-SW		X

Table 19: Mapping between additional threats and objectives for the SW isolation package

In the following, the justification of the coverage of the policies by the security objectives is given.

The OSP P.Access-Ctrl-to-TSF is addressed by O.TSF-Access, which requires the TOE to control the access to security services and hardware resources. In addition, the TOE shall only allow defined operations on TSF data.

The OSP P.Access-Ctrl-to-Composite-SW is addressed by O.Mem-Access, which requires the TOE to control access to memory for each application.

7.5.3 Extended Component Definition

This package does not define additional extended components.

7.5.4 IT Security Requirements

7.5.4.1 SFRs for the TOE

The TOE shall meet the requirement "Management of TSF data (FMT_MTD.1)" as specified below.

FMT_MTD.1/SWIso	**Management of TSF data**
Hierarchical to:	No other components.
Dependencies:	FMT_SMR.1 Security roles
	FMT_SMF.1 Specification of Management Functions
FMT_MTD.1.1/SWIso	The TSF shall restrict the ability to [selection: *change_default, query, modify, delete, clear, [assignment: other operations]*] the [assignment: *list of TSF data*] to FW and SW enforcing TSF[37].

[37] [assignment: the authorised identified roles]

Refinement:	**The assignment "the authorised identified roles" is limited to the FW and SW of the 3S. Only FW and SW shall be able to process keys and attributes enforcing the protection and use of TSF data.**

Application Note 66. The ST writer shall define the operations that are allowed on specific TSF data by dedicated functions of the FW and SW that is part of the TOE. This may comprise the use and management of keys for the verification of software downloads, Root of Trust, access permission to manufacturer data, permission to security services and hardware resources by Composite Software.

The TOE shall meet the requirement "Specification of Management Functions (FMT_SMF.1)" as specified below.

FMT_SMF.1/SW_TSF	**Specification of Management Functions**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1/SW_TSF	The TSF shall be capable of performing the following management functions: [assignment: *list of management functions to be provided by the TSF*].

Application Note 67. The ST writer shall iterate the SFRs for access control, depending on the functionality provided by the TOE. E.g., different access control policies may be implemented for memory and hardware resources. The iteration is recommended if the two access control policies are based on different security mechanisms implemented in the TOE. The SFR for the "Specification of Management Functions" may be the same if different access control policies are defined.

Application Note 68. The access control to memory may be defined, based on memory addresses or memory pages, depending on the implementation of the TOE.

The TOE shall meet the requirement "Complete access control (FDP_ACC.2)" as specified below.

FDP_ACC.2/SWIso	**Complete access control**
Hierarchical to:	FDP_ACC.1 Subset access control
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.2.1/SWIso	The TSF shall enforce the [assignment: *access control SFP*] on [assignment: *list of subjects and objects*] and all operations among subjects and objects covered by the SFP.
FDP_ACC.2.2/SWIso	The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

Application Note 69. The "list of subjects" in the SFR is determined by the software running in privileged or unprivileged operation mode. The TOE may implement more than two operation modes. The "list of objects" may include code and data.

The TOE shall meet the requirement “Security attribute based access control (FDP_ACF.1)” as specified below.

FDP_ACF.1/SWIso	**Security attribute based access control**
Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control
	FMT_MSA.3 Static attribute initialisation
FDP_ACF.1.1/SWIso	The TSF shall enforce the [assignment: *access control SFP*] to objects, based on the following: [assignment: *list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes*].

Application Note 70. The list of security attributes shall ensure that the separation of different applications can be enforced.

FDP_ACF.1.2/SWIso	The TSF shall enforce the following rules to determine whether an operation among controlled subjects and controlled objects is allowed: [assignment: *rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects*].
FDP_ACF.1.3/SWIso	The TSF shall explicitly authorise access of subjects to objects, based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly authorise access of subjects to objects*].
FDP_ACF.1.4/SWIso	The TSF shall explicitly deny access of subjects to objects, based on the following additional rules: [assignment: *rules, based on security attributes, that explicitly deny access of subjects to objects*].

The TOE shall meet the requirement “Static attribute initialisation (FMT_MSA.3)” as specified below.

FMT_MSA.3/SWIso	**Static attribute initialisation**
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes
	FMT_SMR.1 Security roles
FMT_MSA.3.1/SWIso	The TSF shall enforce the [assignment: *access control SFP, information flow control SFP*] to provide [selection, *choose one of: restrictive, permissive, [assignment: other property]*] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2/SWIso	The TSF shall allow the [assignment: *the authorised identified roles*] to specify alternative initial values to override the default values when an object or information is created.

The TOE shall meet the requirement “Management of security attributes (FMT_MSA.1)” as specified below.

FMT_MSA.1/SWIso	**Management of security attributes**
Hierarchical to:	No other components.

Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1/SWIso	The TSF shall enforce the [assignment: *access control SFP(s), information flow control SFP(s)*] to restrict the ability to [selection: *change_default, query, modify, delete, [assignment: other operations]*] the security attributes [assignment: *list of security attributes*] to [assignment: *the authorised identified roles*].

The TOE shall meet the requirement "Specification of Management Functions (FMT_SMF.1)" as specified below.

FMT_SMF.1/SWIso	**Specification of Management Functions**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FMT_SMF.1.1/SWIso	The TSF shall be capable of performing the following management functions: [assignment: *list of management functions to be provided by the TSF*].

7.5.4.2 Rationale for the SFRs

Table 20 provides an overview, how the SFRs are combined to meet the security objectives. The justification for each objective follows the table.

Objective	TOE Security Functional and Assurance Requirements
O.TSF-Access	FMT_MTD.1/SWIso Management of TSF data FMT_SMF.1/SW_TSF Specification of Management Functions
O.Mem-Access	FDP_ACC.2/SWIso Complete access control FDP_ACF.1/SWIso Security attribute based access control FMT_MSA.3/SWIso Static attribute initialisation FMT_MSA.1/SWIso Management of security attributes FMT_SMF.1/SWIso Specification of Management Functions

Table 20: Mapping between Objectives and SFRs for the Software Isolation Package

The SFR FMT_MTD.1/SWIso and FMT_SMF.1//SW_TSF defined in this Protection Profile support the objective O.TSF-Access.

The justification related to the security objective "Access and Operation control on TSF data (O.TSF-Access)" is as follows:

The SFR FMT_MTD.1/SWIso ensures that only defined operations are performed by operations of the FW and SW as part of the TOE. FMT_SMF.1//SW_TSF allow only defined and controlled modifications of the TSF data and the associated operations. Therefore, these SFRs support the objective.

The SFR FDP_ACC.2/SWIso, FDP_ACF.1/SWIso, FMT_MSA.3/SWIso, FMT_MSA.1/SWIso and FMT_SMF.1//SWIso defined in this Protection Profile support the objective O.Mem-Access.

The justification related to the security objective “Access control on memory and hardware resources (O.Mem-Access)” is as follows:

The SFR FDP_ACC.2/SWIso defines the access control policy that is implemented by FDP_ACF.1/SWIso. FDP_ACF.1/SWIso ensures that only defined operations can be performed on code and data stored in the memories and that access is limited for each application. FMT_MSA.3/SWIso and FMT_MSA.1/SWIso define the initialisation and the management of the security attributed used by the access control policy. FMT_SMF.1/SWIso allow only defined and controlled modifications of the access control policy. Therefore, these SFRs support the objective.

7.5.4.3 Dependencies of the SFRs

Requirement	No dependency	Satisfied Dependencies
FMT_MTD.1/SWIso	FMT_SMF.1 FMT_SMR.1	FMT_SMF.1/SW_TSF not satisfied, see the rationale below the table
FMT_SMF.1/SW_TSF	No dependency	
FDP_ACC.2/SWIso	FDP_ACF.1	FDP_ACF.1/SWIso
FDP_ACF.1/SWIso	FDP_ACC.1 FMT_MSA.3	FDP_ACC.2/SWIso (because it is hierarchical to FDP_ACC.1) FMT_MSA.3/SWIso
FMT_MSA.3/SWIso	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/SWIso not satisfied, see the rationale below the table
FMT_MSA.1/SWIso	FDP_ACC.1 or FDP_IFC.1 FMT_SMF.1 FMT_SMR.1	FDP_ACC.1/SWIso FMT_SMF.1/SWIso not satisfied, see the rationale below the table
FMT_SMF.1/SWIso	No dependency	

Table 21: Overview of SFR dependencies for the Software Isolation Package

FMT_SMR.1 requires the definition of security roles. This PP leave the decision to define this SFR and its dependencies to the ST author. In case the TOE does not implement different roles the definition of these SFR is left to the composite software.

8 References and Acronyms

8.1 References

8.1.1 Criteria

[1] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Version 3.1, Revision 5, April 2017, CCMB-2017-04-001

[2] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-002

[3] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Version 3.1, Revision 5, April 2017, CCMB-2017-04-003

[4] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Version 3.1, Revision 5, April 2017, CCMB-2017-04-004

[5] Evaluation of random number generators, Bundesamt für Sicherheit in der Informationstechnik, Version 0.1, March 2013

[6] A proposal for: Functionality classes for random number generators, Killmann, W. Schindler, Version 2.0, September 18, 2011

8.1.2 Scheme documents

[7] SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms 1.2, January 2020, see (https://www.sogis.eu/documents/cc/crypto/SOGIS-Agreed-Cryptographic-Mechanisms-1.2.pdf)

8.1.3 Protection Profiles

[8] Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, 13.01.2014, registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014

8.1.4 Specifications

[9] NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Rev. 1, June 2015

[10] NIST Special Publication 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation, January 2018

8.2 Acronyms

Acronym	Definition
CC	Common Criteria
3S	Security Sub-System
EAL	Evaluation Assurance Level
IC	Integrated Circuit
PP	Protection Profile
RNG	Random Number Generator
SOC	System-On-a-Chip
ST	Security Target
TOE	Target Of Evaluation
TSF	TOE Security Functionality

9 Appendix

9.1 Details of the Conformance Rationale

This section includes the detail mapping showing the conformance between this Protection Profile and the Protection Profile BSI-CC-PP-0084-2014.

The tables in this section show the conformance between the security problem definition, the security objectives and the security requirements defined in BSI-CC-PP-0084-2014 and in this PP.

The threats in this PP are a superset of the threats in BSI-CC-PP-0084-2014 [8], to which conformance is claimed, as described in the following table:

Threat in PP0084	Threat in this PP	Rationale
T.Leak-Inherent	T.Leak-Inherent	The threat addresses the same attacker, the same assets and the same adverse action.
T.Phys-Probing	T.Phys-Probing	The threat addresses the same attacker, the same assets and the same adverse action.
T.Malfunction	T.Malfunction	The threat addresses the same attacker, the same assets. The adverse actions are extended in this 3S PP to address the increase software component including additional driver and interfaces due to the integration.
T.Phys-Manipulation	T.Phys-Manipulation	The threat addresses the same attacker, the same assets and the same adverse action.
T.Leak-Forced	T.Leak-Forced	The threat addresses the same attacker, the same assets and the same adverse action. Further on, the integration of the platform in the SoC is addressed in this 3S PP.
T.Abuse-Func	T.Abuse-Func	The threat addresses the same attacker, the same assets and the same adverse action.
T.RND	T.RND	The threat addresses the same attacker, the same assets and the same adverse action.
	T.Insecure-State	This threat was added in the 3S PP to address threats on the additional Root of Trust functionality and the integration of the 3S in the SoC.

Table 22: Comparison between threats in [8] and this PP

The OSP in this 3S PP is taken over and renamed compared to the OSPs in BSI-CC-PP-0084-2014 [8], to which conformance is claimed, as described in the following table:

OSP in PP0084	OSP in this PP	Rationale
P.Process-TOE	P.Gen-Unique-ID	The policy is taken over with the extension that the unique identification

Table 23: Comparison between OSPs in [8] and this PP

The assumptions in this 3S PP are slightly adapted compared to the assumptions in BSI-CC-PP-0084-2014 [8], to which conformance is claimed. In addition, two assumptions are added. These assumptions are not assigned to the usage phase (Phase 7) and do not mitigate any of the defined threats or OSPs.

Assumptions in PP0084	Assumptions in this PP	Rationale
A.Process-Sec-IC	A.Process-Sec-IC	The assumption is taken over with the same scope.
A.Resp-Appl	A.Resp-Appl	The assumption is taken over with the same scope.
	A.Packaging-Requirement	The packing specification for the SoC may take the requirements from the 3S integration into account. Packing requirements are not taken into account for the Security IC.

Table 24: Comparison between assumptions in [8] and this PP

The Security Objectives in this 3S PP are extended compared to the Security Objectives in BSI-CC-PP-0084-2014 [8], to which conformance is claimed, as described in the following table:

Security Objectives in PP0084	Security Objectives in this PP	Rationale
O.Leak Inherent	O.Leak Inherent	The Security Objective is taken over with the same scope.
O.Phys Probing	O.Phys Probing	The Security Objective is taken over with the same scope.
O.Malfunction	O.Malfunction	The Security Objective is taken over and extended regarding the security requirements on software.
O.Phys Manipulation	O.Phys Manipulation	The Security Objective is taken over with the same scope.
O.Leak Forced	O.Leak Forced	The Security Objective is taken over with the same scope.
O.Abuse Func	O.Abuse Func	The Security Objective is taken over with the same scope.
O.RND	O.RND	The Security Objective is taken over with the same scope.
	O.Secure-State	Additional Security Objective for the secure start-up and the additional Root of Trust functionality
O.Identification	O.Identification	The Security Objective is taken over with the same scope.
OE.Resp Appl	OE.Resp-Appl	The Security Objective is taken over with the same scope.
OE.Process-Sec-IC	OE.Process-Sec-IC	The Security Objective is taken over with the same scope.

Security Objectives in PP0084	Security Objectives in this PP	Rationale
	OE.Packaging-Requirement	Additional objective for the environment, because the packaging is not defined in BSI-CC-PP-0084-2014.

Table 25: Comparison between Security Objectives in [8] and this PP

The security requirements in this PP are a superset of the security requirements in the PP [8] (BSI-CC-PP-0084-2014), to which conformance is claimed. The security objectives of the TOE are mapped to the same SFRs in both Protection Profiles with the following differences:

Security Objectives in PP0084	Security Objectives in this PP	Rationale
FDP_ITT.1 "Basic internal transfer protection" FPT_ITT.1 "Basic internal TSF data transfer protection" FDP_IFC.1 "Subset information flow control"	FPT_EMS.1 **"**Emanation of TSF and user data"	For "O.Leak-Inherent" three SFR are replaced by a new dedicated SFR addressing leakage protection. The SFR has already been used in other PPs and is commonly defined. The SFR has the same scope as the three SFR including the leakage protection of TSF data and User Data.
	FPT_INI.1 "TSF Initialisation"	The protection against "O.Malfunction" is extended by an SFR for the secure TSF initialization. This is an extension of the security functionality defined in PP0084. The SFR also supports "O.Identification".
FMT_LIM.1 "Limited capabilities" FMT_LIM.2 "Limited availability"	FMT_LIM.1/Test Limited capabilities FMT_LIM.2/Test Limited availability FMT_LIM.1/Debug Limited capabilities FMT_LIM.2/Debug Limited availability	The aspect of abuse was split to address the potentially extended attack surface with separate test and debug interfaces. This is an extension of the security functionality defined in PP0084.

Table 26: Comparison between the SFRs in [8] and this PP.

9.2 Informative Guidance for the Definition of the SFR for the RNG

This chapter provides informative examples of security requirements defined for RNG in some national certification schemes and how to perform the operations in the SFR FCS_RNG.1.

9.2.1 Bundesamt für Sicherheit in der Informationstechnik (BSI) Scheme

The Bundesamt für Sicherheit in der Informationstechnik (BSI) published mandatory evaluation requirements for the German Common Criteria certification scheme [5]. These documents describe predefined classes PTG.2, PTG.3 and DRG.4 of random number generators (cf. [6]) appropriate for the TOE of this protection profile.

The most commonly used pre-defined class is the physical random number generator PTG.2. The SFR "Random Number Generation – PTG.2 (FCS_RNG.1/PTG.2)" can be defined according the following proposal (without performed operation, cf. application note).

FCS_RNG.1/PTG.2	**Random number generation – PTG.2**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1/PTG.2	The TSF shall provide a physical[38] 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 [selection: *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, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy*].
(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 shall 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 [selection: *externally, at regular intervals, continuously, applied upon specified internal events*]. 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[39].
FCS_RNG.1.2/PTG.2	The TSF shall provide [selection: bits, octets of bits, numbers [assignment: format of the numbers]] that meet the following:
(PTG.2.6)	Test procedure A [assignment: *additional standard test* suites] 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 0.997[40].

Application Note 71. The ST writer shall perform the missing operations appropriate for cryptographic application of the random numbers in the elements FCS_RNG.1.1 and FCS_RNG_1.2. The ST writer shall perform the selections for specification of the security capabilities provided by the random number generator of the TOE. The evaluation of the random number generator shall follow a recognised methodology (e.g., AIS31, cf. [5]).

[38] [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]
[39] [assignment: *list of security capabilities*]
[40] [assignment: *a defined quality metric*]

9.2.2 National Institute of Standards and Technology (NIST) Scheme

The following two informative examples show how FCS_RNG.1 may be used for SFR of physical RNG and hybrid deterministic RNG meeting the security requirements and designs of cryptographic post-processing in [9] and [10].

The National Institute of Standards and Technology (NIST) published NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Rev. 1, June 2015 [9] and NIST Special Publication 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation, January 2018 [10]. The draft recommendation for entropy sources [10] describes security requirements and test procedures that may be applied to the entropy source of a deterministic random number generator or a physical random number generator of the TOE. [9] defines hybrid deterministic RNG designs. Note [9] is currently under construction and only the designs based on block ciphers and hash functions should be used.

If the TOE shall implement a physical random number generator as entropy source compliant to [10] the ST writer may define an SFR "Random Number Generation – ES (FCS_RNG.1/ES)", as follows:

FCS_RNG.1/ES	**Random number generation**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1/ES	The TSF shall provide a *physical*[41] random number generator that implements the following:
(ES.1)	Failure or severe degradation of the noise source shall be detectable.
(ES.2)	Continuous tests or other mechanisms in the entropy source shall protect against producing output during malfunctions.
(ES.3)	[assignment: *list of additional security capabilities*][42].
FCS_RNG.1.2/ES	The TSF shall provide [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*] that meet the following:
(ES.4)	each output bit is independent of all other output bits,
(ES.5)	[selection:
(ES.5a)	full entropy output,
(ES.5b)	[assignment: bias and entropy rate of the output]][43].

The clause (ES.3) may describe conditioning components implementing NIST approved or non-approved cryptographic functions, which are optional in [10]. A full entropy source provides bit strings output containing at least $(1-\varepsilon)n$ bits entropy, where n is the length of each output string and $0 \leq \varepsilon \leq 2^{-64}$.

If the TOE shall implement hybrid random number generator of the TOE complying to [9] seeded by a physical random number generator as entropy source described above the ST writer may define an SFR "Random Number Generation – Hybrid deterministic RNG (FCS_RNG.1/HD)", as follows:

FCS_RNG.1/HD	**Random number generation – Hybrid deterministic RNG**
Hierarchical to:	No other components.
Dependencies:	No dependencies.

[41] [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]
[42] [assignment: *list of security capabilities*]
[43] ([assignment: *a defined quality metric*]

FCS_RNG.1.1/HD	The TSF shall provide a *hybrid deterministic*[44] random number generator that implements: [selection: *CTR_DRBG, Hash_DRBG, HMAC_DRBG*] as defined in NIST Special Publication 800-90A [9][45].
FCS_RNG.1.2/HD	The TSF shall provide [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*] that meet [assignment: *security bits*][46].

For details of the security capabilities and the security bits as quality metric of the random number output, see NIST Special Publication 800-90A [9].

[44] [selection: *physical, non-physical true, deterministic, hybrid physical, hybrid deterministic*]
[45] [assignment: *list of security capabilities*]
[46] [assignment: *a defined quality metric*]