EUROSMART
The Voice of the Digital Security Industry

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

Version 1.8

26 October 2023

Sponsored by:

AMD

Qualcomm

winbond

Registered and Certified by
Federal Office for Information Security (BSI)
under the reference BSI-CC-PP-0117-V2

Version	Date	Changes
1.8	26.10.2023	Minor changes to address BSI comment.

Table of Contents

1 PP Introduction

1.1 PP Reference

Title	Secure Sub-System in System-on-Chip (3S in SoC)
Version:	1.8
Date:	26 October 2023
Developer:	Eurosmart
Technical Editor:	Deutsche Telekom Security GmbH
Certification Body:	Bundesamt für Sicherheit in der Informationstechnik (BSI)
Certification ID:	BSI-CC-PP-0117-V2

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. Also guidance for the integration of the 3S and the reporting of any suspected security flaws in the TOE are part of the TOE.

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

[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.

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

The 3S is a physically-fixed design defined either as a hard macro (e.g., a GDSII file) and/or as a programmable logic (PL) macro (a bitstream used to configure a Field Programmable Gate Array (FPGA)). 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.

Application Note 1. The PL Macro security solution should be detailed, including additional SFRs if required. The Security Target (ST) author shall supplement the description, based on the specific implementation.

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 2. 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 as needed.

Application Note 3. 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 4. 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 in functional terms: The FW cannot be executed before the hardware is powered on and the SW is 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.

Although the bootup order in functional terms is the indicated above, it might be required that a complete initialization of the FW and/or the SW is required in order to finish the secure configuration of the HW. It is possible that the FW and/or the SW need to set some security parameters of the HW and that are required to the securely initialization.

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.

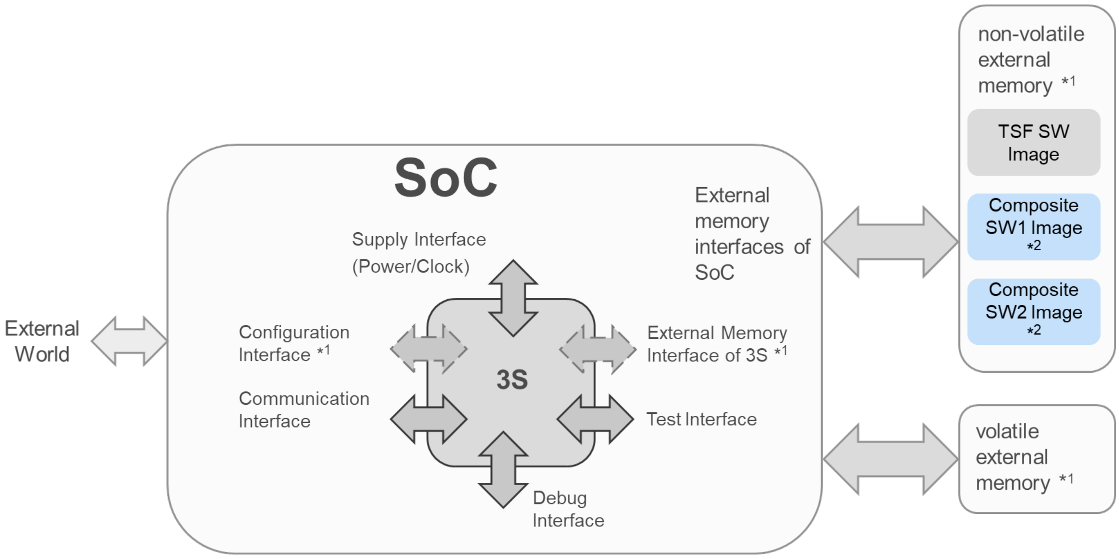

non-volatile external memory *1
TSF SW Image
Composite SW1 Image *2
Composite SW2 Image *2
SoC
External memory interfaces of SoC
Supply Interface (Power/Clock)
External World
Configuration Interface *1
External Memory Interface of 3S *1
3S
Communication Interface
Test Interface
Debug Interface
volatile external memory *1

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 5. 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 circuitry controlling these interfaces shall be completely included in the 3S.

Application Note 6. 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 7. 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 sections 7.1 and 7.2 are defined as separate packages; for details, 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.

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 (RoT)
- 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.

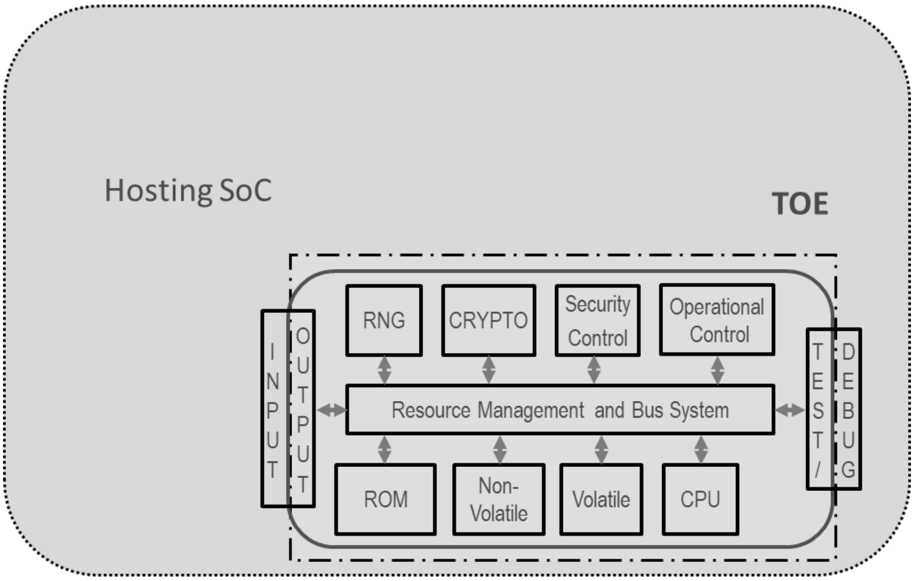


Figure 2: All components are integrated inside the 3S

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.

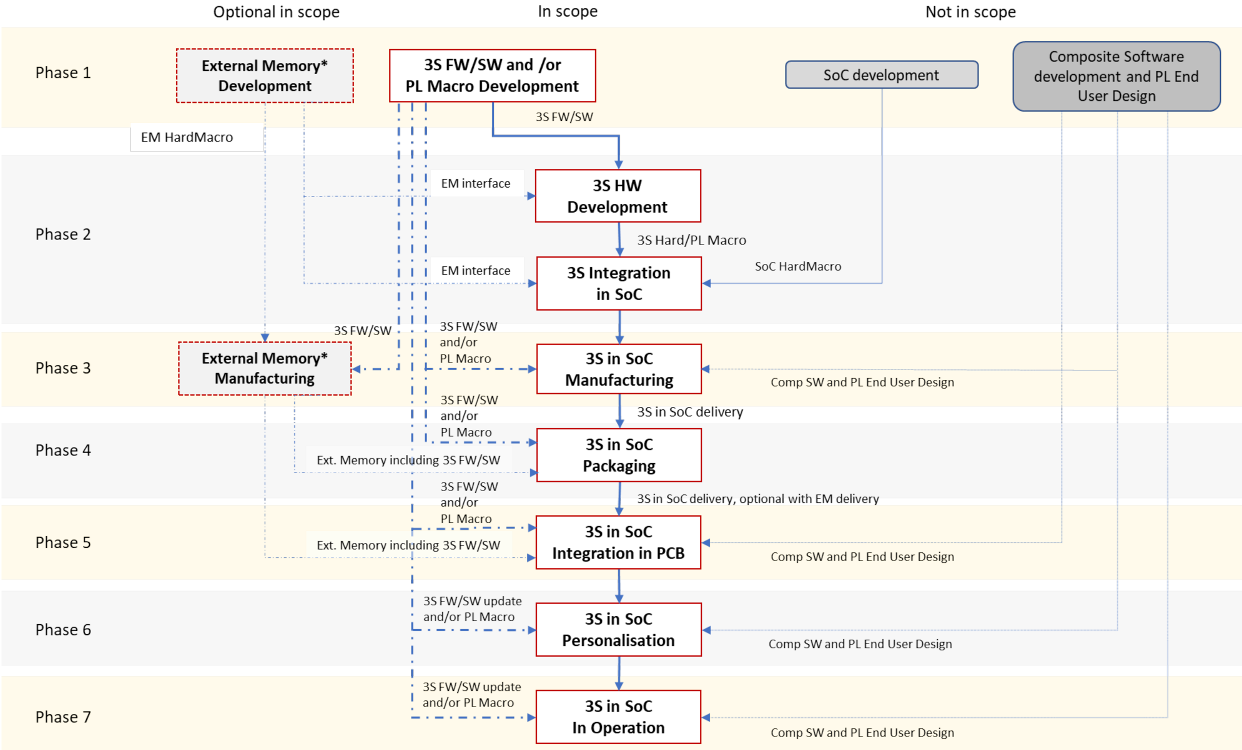
The complex hardware and software development process of System on Chips including a 3S can be split into seven generic phases. The form factor and the integration of the SoC are not standardised. Therefore, the life cycle can depend on the intended usage of the Composite Product. This can comprise the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


Optional in scope
In scope
Not in scope
Phase 1
Phase 2
Phase 3
Phase 4
Phase 5
Phase 6
Phase 7
External Memory* Development
3S FW/SW and /or PL Macro Development
SoC development
Composite Software development and PL End User Design
3S FW/SW
EM HardMacro
EM interface
3S HW Development
3S Hard/PL Macro
EM interface
3S Integration in SoC
SoC HardMacro
3S FW/SW
3S FW/SW and/or PL Macro
External Memory* Manufacturing
3S in SoC Manufacturing
Comp SW and PL End User Design
3S FW/SW and/or PL Macro
3S in SoC delivery
3S in SoC Packaging
Ext. Memory including 3S FW/SW
3S FW/SW and/or PL Macro
3S in SoC delivery, optional with EM delivery
3S in SoC Integration in PCB
Ext. Memory including 3S FW/SW
Comp SW and PL End User Design
3S FW/SW update and/or PL Macro
3S in SoC Personalisation
Comp SW and PL End User Design
3S FW/SW update and/or PL Macro
3S in SoC In Operation
Comp SW and PL End User Design

* 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 TOE 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 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 and personalisation 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 personalisation are completed and the self-protection of the 3S is completely enabled. The evaluation shall include all manufacturing steps, 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 personalisation 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 personalisation 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 be considered as a delivery item only at the end of Phase 4, if the trimming, initialisation and personalisation are completed and the self-protection of the 3S is completely enabled. The evaluation shall include all manufacturing steps, which require 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 personalisation. 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:

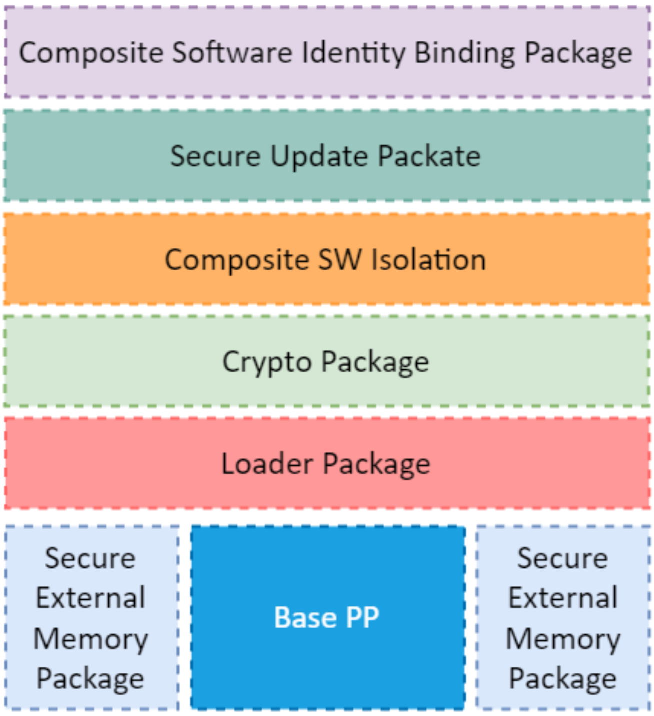

Composite Software Identity Binding Package
Secure Update Packate
Composite SW Isolation
Crypto Package
Loader Package
Secure External Memory Package
Base PP
Secure External Memory Package

Figure 4: Package structure of this Protection Profile

Package Name	Package Purpose	Reference	Relationship
Base PP		Sections 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
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
Secure Update Package	The package provides the secure update feature for the TOE in order to severely reduce the risk of exploitation of a potential vulnerability.	Section 7.6	Optional
Composite Software identity binding with asymmetric cryptographic key Package	The package provides the process for key provisioning that occurs during 3S in SoC packaging, providing asymmetric cryptography key material to the 3S in SoC before 3S in SoC delivery.	Section 7.7	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), CC:2022, Revision 1:

Conformance of this PP with respect to CC Part 1 (Introduction and general model), see [3].

Conformance of this PP with respect to CC Part 2 (Security Functional components) is CC Part 2 extended, see [4].

Conformance of this PP with respect to CC Part 3 (Security Assurance components) is CC Part 3 conformant, see [5].

Conformance of this PP with respect to CC Part 4 (Framework for the specification of evaluation methods and activities) is CC Part 4 conformant, see [1]. However, CC Part 4 is not used in this PP.

Conformance of this PP with respect to CC Part 5 (Pre-defined packages of security requirements) is CC Part 5 conformant, see [2].

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 [12]. The conformance claim applies to the base Protection Profile without the additional functional packages.

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, AVA_VAN.5, ALC_DVS.2[2] and ALC_FLR.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.

Several SFRs from [12] have been adapted in this PP as BSI-CC-PP-0084-2014 claims conformance to CC Version 3.1, Revision 5, and this PP claims conformance to CC:2022, Revision 1. CC:2022, Revision

[2] The assurance components ATE_DPT.2, AVA_VAN.5 and ALC_DVS.2 represent the augmentations selected in PP0084.

1 has included new SFRs and has modified the instantiation of some SFRs. Section 9.3 provides a summary of the SFRs included in this PP that have been modified according to CC:2022.

2.5 Conformance Statement

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

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.

[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 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.

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.

As stated in section 1.2.2, this Protection Profile requires the TOE to provide generation of random numbers security service by means of a physical Random Number Generator. Section 7.4 provides a general optional package for cryptographic services.

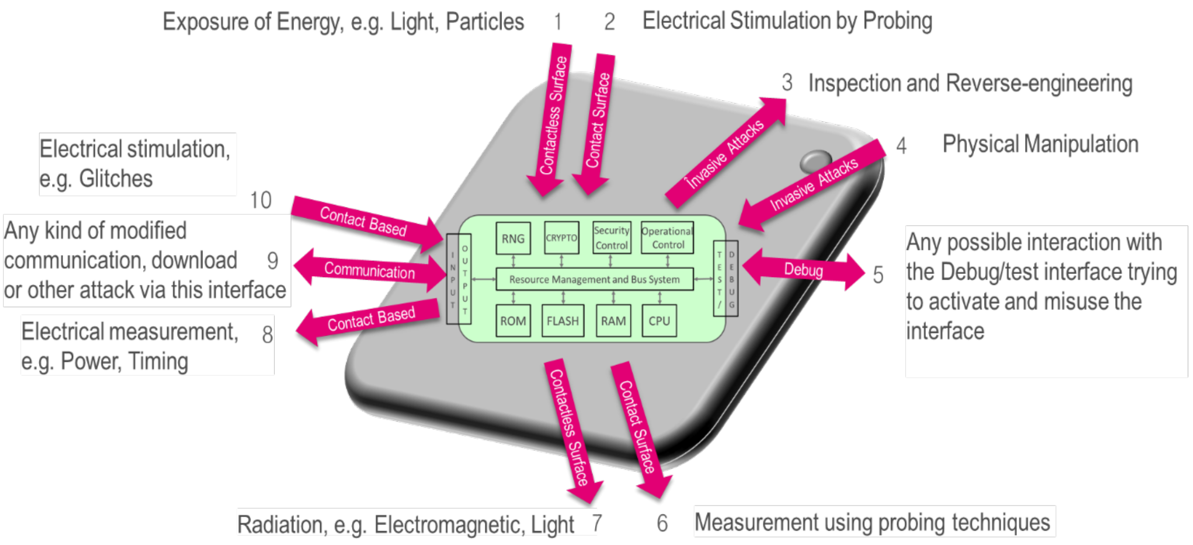
According to this Protection Profile there is the following high-level security concern related to Random Number Generator security service:

SC4: deficiency of random numbers.

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.



Exposure of Energy, e.g. Light, Particles
1
2
Electrical Stimulation by Probing
3
Inspection and Reverse-engineering
4
Physical Manipulation
Electrical stimulation, e.g. Glitches
10
Any kind of modified communication, download or other attack via this interface
9
Electrical measurement, e.g. Power, Timing
8
Any possible interaction with the Debug/test interface trying to activate and misuse the interface
5
Contactless Surface
Contact Surface
Invasive Attacks
Invasive Attacks
Contact Based
Communication
Contact Based
Debug
INPUT
OUTPUT
RNG
CRYPTO
Security Control
Operational Control
Resource Management and Bus System
ROM
FLASH
RAM
CPU
TEST/
DEBUG
Contactless Surface
Contact Surface
Radiation, e.g. Electromagnetic, Light
7
6
Measurement using probing techniques

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 information , as user data or TSF data, which is leaked from the TOE and/or the SoC interfaces while being stored and/or processed by the TOE.

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.

An attacker may also predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided.

This threat addresses the analysis of random numbers generated 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 “Identification of each TOE instance (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 that the 3S does not distinguish between user data that are publicly known or kept confidential. Therefore, the 3S shall protect the user data in integrity and in confidentiality if stored in protected memory areas, unless the Composite Software chooses to disclose or modify this user data. 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 will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have a sufficient entropy.
The TOE will ensure that no information about the generated random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.
The TOE shall detect and/or prevent manipulation or influence of the entropy source to ensure cryptographic quality of random number generation.

Application Note 22. If the TOE provides further security services, this may result in having additional security objectives in the Security Target. Section 7.4 provides package for additional security services the TOE may provide.

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 that, 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 23. 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 24. 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-Manipulation				X									
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 assumption 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 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 Audit Data Storage
1

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*].

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/Env)" as specified below.

FRU_FLT.2/Env	**Limited fault tolerance**
Hierarchical to:	FRU_FLT.1 Degraded fault tolerance
Dependencies:	FPT_FLS.1 Failure with preservation of secure state.
FRU_FLT.2.1/Env	The TSF shall ensure the operation of all the TOE's capabilities when the following failures occur: exposure to operating conditions or usage conditions out of range, which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1/Env)[4].
Refinement:	**The term "failure" above means "circumstances". The TOE prevents failures for the "circumstances" defined above.**

Application Note 25. 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.

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

The TOE shall meet the requirement “Limited fault tolerance (FRU_FLT.2/Log)” as specified below.

FRU_FLT.2/Log	**Limited fault tolerance**
Hierarchical to:	FRU_FLT.1 Degraded fault tolerance
Dependencies:	FPT_FLS.1 Failure with preservation of secure state.
FRU_FLT.2.1/Log	The TSF shall ensure the operation of all the TOE’s capabilities when the following failures occur: abnormal interface behaviour and/or protocol parameters or protocol sequences that can be tolerated and that are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1/Log)[5].
Refinement:	**The term “failure” above means “circumstances”. The TOE prevents failures for the “circumstances” defined above.**

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

FPT_FLS.1/Env	**Failure with preservation of secure state**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1/Env	The TSF shall preserve a secure state when the following types of failures occur: exposure to operating conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2/Env) and where, therefore, a malfunction could occur[6].
Refinement:	**The term “failure” above also covers “circumstances”. The TOE prevents failures for the “circumstances” defined above.**

Application Note 26. The Security Target shall describe the secure state in case the operating conditions cannot be tolerated. The author of the Security Target should clearly define the secure state give a rationale why the defined state is secure.

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

FPT_FLS.1/Log	**Failure with preservation of secure state**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_FLS.1.1/Log	The TSF shall preserve a secure state when the following types of failures occur: exposure to abnormal interface behaviour and/or protocol parameters or protocol sequences which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2/Log) and where, therefore, a malfunction could occur[7].
Refinement:	**The term “failure” above also covers “circumstances”. The TOE prevents failures for the “circumstances” defined above.**

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

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

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

Application Note 27. The Security Target shall describe the secure state in case of abnormal interface behaviour and/or protocol parameters or protocol sequences cannot be tolerated. The author of the Security Target should clearly define the secure state give a rationale why the defined state is secure.

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 both iterations of 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/Test)” 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 limit its 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 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/Test)” 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.

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

FMT_LIM.2.1/Test 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 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[9].

The TOE shall meet the requirement “Limited capabilities (FMT_LIM.1/Debug)” 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 limit its 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[10].

The TOE shall meet the requirement “Limited availability (FMT_LIM.2/Debug)” 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 in a manner that limits its 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 substantial information about construction of TSF to be gathered which may enable other attacks[11].

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.

[9] [assignment: *Limited capability and availability policy*]

[10] [assignment: *Limited capability and availability policy*]

[11] [assignment: *Limited capability and availability policy*]

FDP_SDC.1.1/3S The TSF shall ensure the confidentiality of the following user data: information of the user data[12] while it is stored in the [selection: *temporary memory, persistent memory, any memory*] .

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*].

Refinement: **This SFR applies for internal memory of the 3S.**

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[13] to the TSF[14] 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) 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.

[12] [selection: *all user data, the following user data [assignment: list of user data*]]

[13] [assignment: *physical tampering scenarios*]

[14] [assignment: *list of TSF devices/elements*]

6.1.4 Protection against Leakage

The security functional requirements “Basic internal transfer protection (FDP_ITT.1)” and “Basic internal TSF data transfer protection (FPT_ITT.1)” have been selected to ensure that the TOE must resist leakage attacks (both for user data and TSF data). The corresponding security policy is defined in the security functional requirement “Subset information flow control (FDP_IFC.1)”.

The TOE shall meet the requirement “Basic internal transfer protection (FDP_ITT.1/3S)” as specified below.

FDP_ITT.1/3S	**Basic internal transfer protection**
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_ITT.1.1/3S	The TSF shall enforce the Data Processing Policy[15] to prevent the disclosure[16] of user data when it is transmitted between physically-separated parts of the TOE.
Refinement:	**The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as physically-separated parts of the TOE.**

The TOE shall meet the requirement “Basic internal TSF data transfer protection (FPT_ITT.1/3S)” as specified below.

FPT_ITT.1/3S	**Basic internal TSF data transfer protection**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITT.1.1/3S	The TSF shall protect TSF data from disclosure[17] when it is transmitted between separate parts of the TOE.
Refinement:	**The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic co-processor) are seen as separated parts of the TOE.**

This requirement is equivalent to FDP_ITT.1/3S above but refers to TSF data instead of user data. Therefore, it should be understood as to refer to the same *Data Processing Policy* defined under FDP_IFC.1/3S below.

The TOE shall meet the requirement “ Subset information flow control (FDP_IFC.1/3S)” as specified below:

FDP_IFC.1/3S	**Subset information flow control**
Hierarchical to:	No other components.
Dependencies:	FDP_IFF.1 Simple security attributes

[15] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

[16] [selection: *disclosure, modification, loss of use*]

[17] [selection: *disclosure, modification*]

FDP_IFC.1.1/3S The TSF shall enforce the Data Processing Policy[18] on all confidential data when they are processed or transferred by the TOE or by the Composite Software[19].

The following Security Function Policy (SFP) Data Processing Policy is defined for the requirement "Subset information flow control (FDP_IFC.1/3S)":

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

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[20] with the capability to store [selection: TOE unique identification data, *Initialisation Data, Pre-personalisation Data, [assignment: other data]*][21] in the [assignment: *type of persistent memory*].

Application Note 31. The integrity and uniqueness of the unique identification of the TOE shall be supported by the development, production and test environment.

Application Note 32. 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 shall provide an initialization function which is self-protected for integrity and authenticity.

[18] [assignment: *information flow control SFP*]

[19] [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]

[20] [assignment: *list of subjects*]

[21] [assignment: *list of audit information*]

FPT_INI.1.2 The TOE initialization function shall ensure that certain properties hold on certain elements immediately before establishing the TSF in a secure initial state, as specified in Table 3:

ID	Properties	Elements
1	Correct configuration of[22]	Configurable and/or trimmable security mechanisms and the unique identification[23]
2	Integrity of[24]	Start-up software, correct initialisation of internal keys[25]
3	Correct initialisation of[26]	Internal keys[27]

Table 3: FPT_INI.1.2

FPT_INI.1.3 The TOE initialization function shall detect and respond to errors and failures during initialization such that the TOE [selection: *is halted, successfully completes initialization with* [selection: *reduced functionality, signaling error state*, [*assignment: list of actions*]].

FPT_INI.1.4 The TOE initialization function shall only interact with the TSF in [assignment*: defined methods*] during initialization.

6.1.6 Generation of Random Numbers

The TOE generates random numbers. This family FCS_RNG Generation of random numbers describes the functional requirements for random number generation used for cryptographic purposes.

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, 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 33. 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

[22] [assignment: *property, for instance authenticity, integrity, correct version*]

[23] [assignment: *list of TSF/user firmware, software or data*]

[24] [assignment: *property, for instance authenticity, integrity, correct version*]

[25] [assignment: *list of TSF/user firmware, software or data*]

[26] [assignment: *property, for instance authenticity, integrity, correct version*]

[27] [assignment: *list of TSF/user firmware, software or data*]

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, AVA_VAN.5 and ALC_FLR.2.

The assurance requirements are:

Class ADV: Development

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

Class AGD: Guidance documents

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

Class ALC: Life-cycle support

CM capabilities	(ALC_CMC.4)
CM scope	(ALC_CMS.4)
Delivery	(ALC_DEL.1)
Development security	(ALC_DVS.2)
Flaw remediation	(ALC_FLR.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 34. 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.

6.2.1 Refinements of the TOE Assurance Requirements

The CCDB, the JILWG and the certification bodies publish supporting documents and guidance documents for evaluation and certification of smartcards and similar devices mandatory under CCRA and SOG-IS or the national certification schemes, cf. [13], [14], [15], [16], [17] and [18]. These documents are updated regularly and are valid for the ongoing evaluation in their actual versions.

The following refinements shall support the comparability of evaluations according to this Protection Profile. Where refinements are not needed, some background information based on such documents is provided. In all cases the background information is informative only. The mandatory documents themselves shall be consulted for exact details and overrule the refinements in case of any inconsistency (e.g., due to updates).

Refinements regarding Delivery procedure (ALC_DEL)

Refinement regarding CM scope (ALC_CMS)

Refinement regarding CM capabilities (ALC_CMC)

Refinement regarding Test Coverage (ATE_COV)

Refinement regarding User Guidance (AGD_OPE)

Refinement regarding Preparative User Guidance (AGD_PRE)

The Refinement is identified by bold type. These refinements refer to some keywords within the Security Assurance Requirements that are emphasised by being underlined.

Application Note 35. The refinements as defined below may also be applicable to a hierarchically higher assurance component of the specific family. If a Security Target includes an additional augmentation, the author of the Security Target has to examine that the refinements as defined below are still applicable.

6.2.1.1 Refinements regarding Delivery procedure (ALC_DEL) Introduction

The Common Criteria assurance component of the family ALC_DEL (delivery procedure) refers to the delivery of (i) the TOE or parts of it (ii) to the user or user's site (Developer of the Composite Software or the Composite TOE Manufacturer). The Common Criteria assurance component

ALC_DEL.1 requires procedures and technical measures to detect modifications and prevent any compromise of the Initialisation Data and/or Pre-personalisation Data and/or assigned other data.

In the particular case of a 3S "material and information" than the TOE itself (which by definition includes the necessary guidance) is exchanged with "users". Therefore, considering the definition of the Common Criteria the following refinement is made regarding the items "TOE" and "to the user or user's site":

The following text reflects the requirements of the selected component ALC_DEL.1:

Developer action elements:

ALC_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2D The developer shall use the delivery procedures.

Content and presentation elements:

ALC_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

Evaluator action elements:

ALC_DEL.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Refinement

For delivery of the TOE to the "Composite Product Manufacturer" or "integrated SoC manufacturer " as consumer, all the external interfaces of the TOE Manufacturer have to be taken into account. These are:

- **The interface with the 3S Software Developer (Phase 1) where information about the 3S, development software and/or tools for software development and possible information about mask options are exchanged and the interface with the Phase after TOE Delivery (Phase 4 or 5) where pre-personalisation data, information about tests, and the product in the form of wafers, sawn wafers (dice) or packaged products are exchanged.**

Application Note 36. The consumer in the context of ALC_DEL is the Composite Product Manufacturer to which the TOE as 3S is delivered. The End-consumer is the consumer of the Composite Product which includes the TOE as platform for the Composite Software.

Application Note 37. All identified critical information about the TOE have to be taken into account in order to avoid any tampering with the actual version or substitution of a false version (including unauthorised modification or replacement).

Application Note 38. Depending on whether the TOE comprises programmable non-volatile memory and/or ROM, in addition to TOE pre-personalisation requirements, the TOE SW and/or keys for the authorised personalisation of the programmable non-volatile memory are delivered to the Composite/ integrated Manufacturer.

6.2.1.2 Refinement regarding CM scope (ALC_CMS)

Introduction

The Common Criteria assurance component of the family ALC_CMS (CM scope) refers to the tracking of specific configuration items within the developers configuration management system.

In the particular case of a 3s it is helpful to clarify the scope of the configuration item “TOE implementation representation”:

The following text reflects the requirements of the selected component ALC_CMS.4:

Developer action elements:

ALC_CMS.4.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.4.1C The configuration list includes the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaws reports and resolution status.

ALC_CMS.4.2C The configuration list shall uniquely identify the configuration items.

ALC_CMS.4.3C For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

Evaluator action elements:

ALC_CMS.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Refinement

The “TOE Software” is as user data not part of the TOE but the whole “TOE Software” or part of it may be delivered together with the TOE (as implemented in the ROM or written by the TOE manufacturer in persistent memory). Therefore, the items “TOE SW” or “authentication data” are only relevant for the configuration list as far as the TOE manufacturer can control these items. Since the TOE Software may be developed by another company it is only available in a specific form and is not part of the TOE though delivered together with it. Authentication data may be required for products implementing programmable non-volatile memory to enable the download of software.

CM list should include 3S deliveries from other than the developer, as IP developers.

Background information

Depending on the product type with programmable non-volatile memory and/or ROM the TOE SW and/or authentication data for a secure loader of the programmable non-volatile memory may be considered as part of the TOE implementation representation.

The “TOE implementation representation” within the scope of the CM will include at least:

- logical design data,
- physical design data,
- IC Dedicated Software,
- final physical design data necessary to produce the photomasks, and
- photomasks.

6.2.1.3 Refinement regarding CM capabilities (ALC_CMC)

Introduction

The Common Criteria assurance component of the family ALC_CMC (CM capabilities) refers to the capabilities of a CM system. The component ALC_CMC.4 “Production support, acceptance procedures and automation" refers to "configuration items" and "configuration list" and uses the term "TOE" in addition.

In the particular case of a 3S the scope of "configuration items" and the meaning of "TOE" in this context need to be clarified:

The following text reflects the requirements of the selected component ALC_CMC.4:

Developer action elements:

ALC_CMC.4.1D	The developer shall provide the TOE and a reference for the TOE.
ALC_CMC.4.2D	The developer shall provide the CM documentation.
ALC_CMC.4.3D	The developer shall use a CM system.

Content and presentation elements:

ALC_CMC.4.1C	The TOE shall be labelled with its unique reference.
ALC_CMC.4.2C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ALC_CMC.4.3C	The CM system shall uniquely identify all configuration items.
ALC_CMC.4.4C	The CM system shall provide automated measures such that only authorised changes are made to the configuration items.
ALC_CMC.4.5C	The CM system shall support the production of the TOE by automated means.
ALC_CMC.4.6C	The CM documentation shall include a CM plan.
ALC_CMC.4.7C	The CM plan shall describe how the CM system is used for the development of the TOE.
ALC_CMC.4.8C	The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.
ALC_CMC.4.9C	The evidence shall demonstrate that all configuration items are being maintained under the CM system.
ALC_CMC.4.10C	The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

Evaluator action elements:

ALC_CMC.4.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Refinement

"Configuration items" comprise all items defined and refined under ALC_CMS (see above) to be tracked under CM.

A production control system has to be applied to guarantee the traceability and completeness of different production charges or lots. The number of wafers, dies and chips must be tracked by this system. Appropriate administration procedures have to be provided for managing wafers, dies or complete chips, which are being removed from the production-process in order to verify and to

control predefined quality standards and production parameters. It has to be controlled that these wafers, dies or assembled devices are returned to the same production stage from which they are taken or they have to be securely stored or destroyed otherwise.

6.2.1.4 Refinement regarding Test Coverage (ATE_COV)

Introduction

The Common Criteria assurance component of the family ATE_COV (test coverage) “addresses the extent to which the TSF is tested, and whether the testing is sufficiently extensive to demonstrate that the TSF operates as specified".

The following text reflects the requirements of the selected component ATE_COV.2:

Developer action elements:

ATE_COV.2.1D The developer shall provide an analysis of the test coverage.

Content and presentation elements:

ATE_COV.2.1C The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2C The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

Evaluator action elements:

ATE_COV.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Refinement

The TOE must be tested under different operating conditions within the specified ranges. These conditions include but are not limited to the frequency of the clock, the power supply, and the temperature. This means that “Fault tolerance (FRU_FLT.2)” must be proven for the complete TSF. The tests must also cover functions which may be affected by “ageing”.

The existence and effectiveness of mechanisms against physical attacks (as specified by the functional requirement FPT_PHP.3) cannot be tested in a straightforward way. Instead, the TOE Manufacturer shall provide evidence that the TOE has the particular physical characteristics (especially layout design principles). This can be done by checking the layout (implementation or actual) in an appropriate way. The required evidence pertains to the existence of mechanisms against physical attacks (unless they are obvious).

Background information

The 3S Dedicated Test Software is seen as a “test tool” delivered as part of the TOE. The Test Features, however, do not provide security functionality. Therefore, Test Features need not be covered by the Test Coverage Analysis, but all functions and mechanisms that limit the capability of the functions (cf. FMT_LIM.1) and control access to the functions (cf. FMT_LIM.2) provided by the 3S Dedicated Test Software must be part of the Test Coverage Analysis.

6.2.1.5 Refinement regarding User Guidance (AGD_OPE)

Introduction

The Common Criteria assurance components of the families AGD_OPE (Operational user guidance) and AGD_PRE (Preparative user guidance) “describe all relevant aspects for the secure application of the TOE“.

The Operational User Guidance documents should provide only the information which is necessary for using the TOE. Depending on the recipient of that guidance documentation Operational and Preparative User Guidance can be given in the same document.

After production the TOE is tested where communication is performed by directly contacting the pads that mostly become part of the interface during packaging. Here no guidance document according to Common Criteria class AGD is required (provided that the tests are performed by the TOE Manufacturer). Note that test procedures are described under the Common Criteria assurance component of the family ATE_FUN.

The following text reflects the requirements of the selected component AGD_OPE.1:

Developer action elements:

AGD_OPE.1.1D The developer shall provide the operational user guidance.

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user- accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

AGD_OPE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Refinement

The TOE serves as a platform for the 3S Software. Therefore, the role of the developer of the 3S Software is the main focus of the guidance, see also section 6.2.1.1.

If the TOE provides security functionality which can or need to be administrated (i) by the 3S Software or (ii) if the 3S Software provides additional services (see section 1.2.2), these aspects must be described in Guidance. This may also comprise specific functionality that must be provided by the 3S Software to support the security of the platform and configuration options of the TOE.

Guidance documents must not contain security relevant details which are not necessary for the usage or administration of the security functionality of the TOE.

Background information

Most of the security functionality will already be effective before TOE Delivery. However, guidance to determine the behaviour of security functionality, to disable, to enable or to modify the behaviour of security functionality must be given if a configuration is possible after TOE Delivery (that means either by the Developer of the 3S Software or by the Composite Product Manufacturer). This guidance is delivered by the TOE Manufacturer.

6.2.1.6 Refinement regarding Preparative User Guidance (AGD_PRE)

Introduction

Preparative user guidance is intended to be used by those persons responsible for secure acceptance and installation of the TOE as well as the secure preparation of the operational environment in a correct manner for maximum security.

The following text reflects the requirements of the selected component AGD_PRE.1:

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Content and presentation elements:

AGD_PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with developer's delivery procedures.

AGD_PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

AGD_PRE.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

Refinement

The Family AGD_PRE addresses the activities of the delivery acceptance procedures. For the hardware platform this comprises procedures that can be applied to identify the TOE and

eventually to verify the authenticity of that part of the TOE using e.g. the security functionality provided according to FAU_SAS.1.

The TOE may be configured after production before the Composite Product is delivered to the consumer. In this case, these configuration aspects have to be considered. Differences between the TOE before first use (normally done during wafer test) and Phase 7 must be summarised. Guidance to change that behaviour must exist.

The preparation may include the download of 3S Software if parts of the 3S Software are stored in the programmable non-volatile memory. If the TOE includes software that is delivered separately the preparation includes integration of the 3S Software. The preparation also includes the configuration of the TOE according to the options described in the ST that can be changed after TOE delivery. The guidance documentation shall describe all relevant procedures.

6.2.2 Refinements of the TOE Integration Assurance Requirements

The 3S 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 should be described in the integration guidance.

The refinements ensure that the integration process will be evaluated as part of the TOE evaluation.

The following refinements shall support the comparability of evaluations according to this Protection Profile:

ADV_ARC.1 Architectural design

Refinements related to the integration guidance:

- ADV_ARC.1.4D: 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.1.6C: The rationale shall be at the level of detail of the TOE design and the integration guidance requirements.

- ADV_ARC.1.2E in CEM: The evaluator shall examine the security architecture description to determine that the information provided in the evidence is presented at a level of detail commensurate with the descriptions of the SFR-enforcing abstractions contained in the functional specification and TOE design document. TOE integration guidance should be examined as well.

AGD_OPE.1 Operational user guidance

AGD_PRE.1 Preparative user guidance

Refinements related to the integration guidance:

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

6.3 Security Requirements Rationale

6.3.1 Rationale for the SFRs

Table 4 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	FDP_ITT.1/3S Basic internal transfer protection FPT_ITT.1/3S Basic internal TSF data transfer protection FDP_IFC.1/3S Subset information flow control
O.Phys-Probing	FDP_SDC.1/3S Stored data confidentiality FPT_PHP.3 Resistance to physical attack
O.Malfunction	FRU_FLT.2/Env Limited fault tolerance FPT_FLS.1/Env Failure with preservation of secure state FRU_FLT.2/Log Limited fault tolerance FPT_FLS.1/Log Failure with preservation of secure state Supported by: FPT_INI.1 TSF Initialisation
O.Phys-Manipulation	FDP_SDI.2/3S Stored data integrity monitoring and action FPT_PHP.3 Resistance to physical attack
O.Leak-Forced	FDP_ITT.1/3S Basic internal transfer protection FPT_ITT.1/3S Basic internal TSF data transfer protection FDP_IFC.1/3S Subset information flow control FRU_FLT.2/Env Limited fault tolerance FPT_FLS.1/Env Failure with preservation of secure state FRU_FLT.2/Log Limited fault tolerance FPT_FLS.1/Log Failure with preservation of secure state FPT_PHP.3 Resistance to physical attack
O.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/Env Limited fault tolerance FPT_FLS.1/Env Failure with preservation of secure state FRU_FLT.2/Log Limited fault tolerance FPT_FLS.1/Log Failure with preservation of secure state FDP_SDI.2/3S Stored data integrity monitoring and action FPT_PHP.3 Resistance to physical attack
O.RND	FCS_RNG.1 Random number generation Supported by: FRU_FLT.2/Env Limited fault tolerance FPT_FLS.1/Env Failure with preservation of secure state FDP_ITT.1/3S Basic internal transfer protection FPT_ITT.1/3S Basic internal TSF data transfer protection FDP_IFC.1/3S Subset information flow control FPT_PHP.3 Resistance to physical attack
O.Secure-State	FPT_INI.1 TSF Initialisation Supported by: FRU_FLT.2/Env Limited fault tolerance FPT_FLS.1/Env Failure with preservation of secure state FRU_FLT.2/Log Limited fault tolerance FPT_FLS.1/Log Failure with preservation of secure state FDP_SDI.2/3S Stored data integrity monitoring and action FPT_PHP.3 Resistance to physical attack

Objective	TOE Security Functional and Assurance Requirements
O.Identification	FAU_SAS.1 Audit storage Supported by: FPT_INI.1 TSF Initialisation

Table 4: Security Requirements versus Security Objectives

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

The SFRs FPT_ITT.1/3S and FDP_ITT.1/3S together with the policy statement in FDP_IFC.1/3S 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_ITT.1/3S and FDP_ITT.1/3S together with the policy statement in FDP_IFC.1/3S in conjunction 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.1/3S 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 covers situations where malfunction of the TOE might be caused by the operating conditions of the TOE or abnormal usage of TOE interfaces (while direct manipulation of the TOE is covered by O.Phys-Manipulation). For the operating conditions the security objective covers the following two circumstances: either all operating conditions are inside the tolerated range or at least one of them is outside this range. The second case is covered by FPT_FLS.1/Env, because it states that a secure state is preserved in this case. The first case is covered by FRU_FLT.2/Env, because it states that the TOE operates correctly under normal (tolerated) conditions. For the abnormal interface behaviour and/or protocol parameters or protocol sequences also two circumstances are covered: Either the interface behaviour can be tolerated as described by FRU_FLT.2/Log or the interface behaviour may cause a mal function and, therefore, shall stop the operation and change to a secure state covered by FPT_FLS.1/Log. The TOE may enter the same a secure state for both iterations of FPT_FLS.1 or defines a secure state for each instance FPT_FLS.1/Env and FPT_FLS.1/Log.

The objective is supported by FPT_INI.1 that ensures the correct initialisation and configuration of the 3S during start-up.

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

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 modifying the interface behaviour or by directly manipulating it) with a second attack step measuring and analysing some output generated by the TOE. The first step is prevented by the SFR FRU_FLT.2/Env, FRU_FLT.1/Log, FPT_FLS.1/Env and FPT_FLS.1/Log for the control of the operating conditions and FPT_PHP.3 that prevent physical modification. Furthermore, the protection against leakage defined by FPT_ITT.1 and FDP_ITT.1 together with the policy statement in FDP_IFC.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/Env and FPT_FLS.1/Env control the operating conditions and prevent malfunctions that may allow to circumvent the control implemented by FMT_LIM.1 and FMT_LIM.2. FRU_FLT.1/Log and FPT_FLS.1/Log control the interface behaviour and prevent malfunctions that may allow to circumvent the control implemented by FMT_LIM.1 and FMT_LIM.2. The SFR FDP_SDI.2/3S 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 4.

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/Env and FPT_FLT.2/Env prevent malfunction of the TOE, based on malicious operating conditions. FPT_PHP.3 prevents physical manipulation and FPT_ITT.1 and FDP_ITT.1 together with the policy statement in FDP_IFC.1 prevent 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 generated random numbers is available to an attacker.

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/Env and FPT_FLS.1/Env controlling the operating conditions and FRU_FLT.1/Log and FPT_FLS.1/Log controlling the interface behaviour prevent malfunctions that may allow to manipulate the secure initialisation. The SFR FDP_SDI.2/3S 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 4.

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 5 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
FDP_ITT.1/3S	FDP_ACC.1 or FDP_IFC.1	Yes by FDP_IFC.1/3S
FDP_IFC.1/3S	FDP_IFF.1	See discussion below
FPT_ITT.1/3S	None	No dependency
FPT_PHP.3	None	No dependency
FDP_SDC.1/3S	None	No dependency
FRU_FLT.2/Env	FPT_FLS.1/Env	Satisfied by FPT_FLS.1/Env
FPT_FLS.1/Env	No dependency	No dependency
FRU_FLT.2/Log	FPT_FLS.1/Log	Satisfied by FPT_FLS.1/Log
FPT_FLS.1/Log	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 5: Overview of SFR dependencies

Part 2 of the Common Criteria defines the dependency of FDP_IFC.1 (information flow control policy statement) on FDP_IFF.1 (Simple security attributes). The specification of FDP_IFF.1 would not capture the nature of the security functional requirement nor add any detail. As stated in the Data Processing Policy referred to in FDP_IFC.1/3S there are no attributes necessary. The security functional requirement for the TOE is sufficiently described using FDP_ITT.1/3S and its Data Processing Policy (FDP_IFC.1/3S).

6.3.3 Rationale for the Assurance Requirements

The assurance level EAL4 and the augmentation with the requirements ALC_DVS.2, ATE_DPT.2, AVA_VAN.5 and ALC_FLR.2 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 ALC_DVS.2 and ALC_FLR.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. In addition the developer needs to implements security flaw reporting procedures for TOE users in order to act appropriately upon reported security flaw and provide corrective fixes.

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.

6.3.3.4 ALC_FLR.2 Flaw reporting procedures

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

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

ALC_FLR.2 has no dependencies.

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

During the operation of the TOE in the field users may identify failures, flaws or abnormal behaviour. An analysis of such events can only be performed by the developer. Therefore, secure continuous operation is supported by a security flaw remediation process implemented by the developer.

7 Definition of Packages

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

Application Note 39. The Protection Profile ensures the uniqueness of each iteration as defined within the different packages. In case, a package needs to be added twice, the author of the Security Target needs to ensure the uniqueness of each SFR definition and the requirements of CC Part 1 [3] regarding the usage of iterations.

Some of the packages have dependencies that need to be considered; for details, 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.
- The 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.

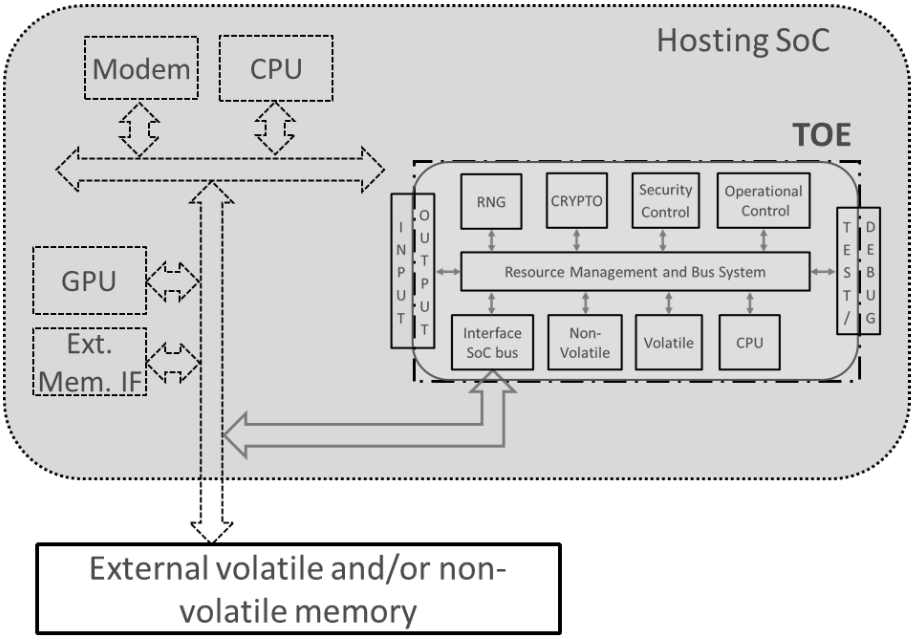


Figure 6: 3S with passive external memory (PM)

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 40. 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).

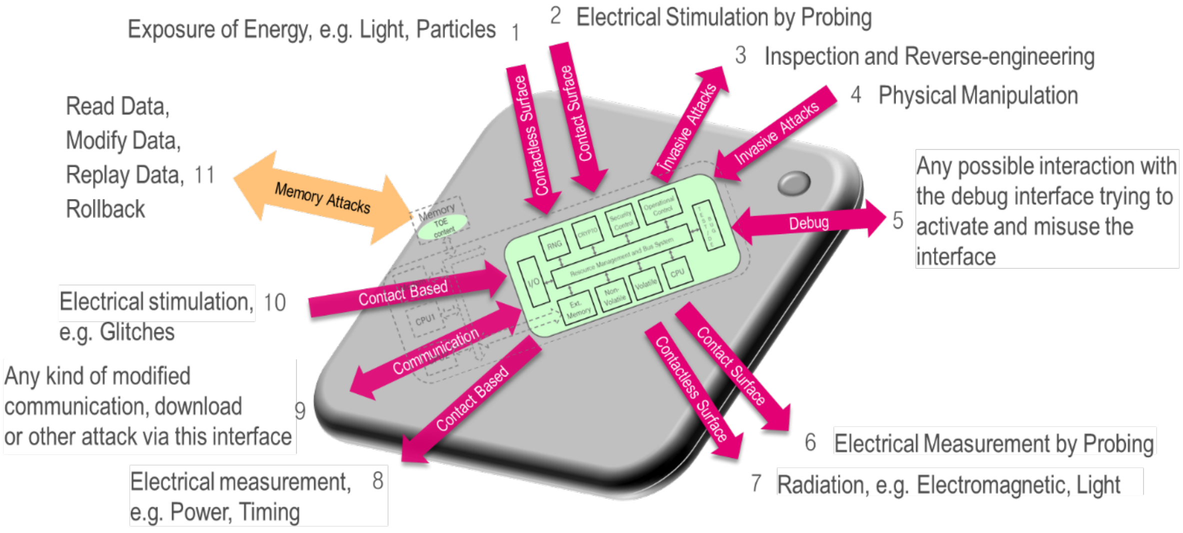

Exposure of Energy, e.g. Light, Particles 1
2 Electrical Stimulation by Probing
3 Inspection and Reverse-engineering
4 Physical Manipulation
Read Data,
Modify Data,
Replay Data, 11
Rollback
Memory Attacks
Contactless Surface
Contact Surface
Invasive Attacks
Invasive Attacks
Any possible interaction with the debug interface trying to activate and misuse the interface
Debug
5
Memory
TOE content
Electrical stimulation, e.g. Glitches 10
Contact Based
CPU1
Communication
Any kind of modified communication, download or other attack via this interface 9
Contact Based
Contactless Surface
Contact Surface
6 Electrical Measurement by Probing
7 Radiation, e.g. Electromagnetic, Light
Electrical measurement, e.g. Power, Timing 8
RNG
CRYPTO
Security Control
Operational Control
I/O
Resource Management and Bus System
Ext Memory
Non-Volatile
Volatile
CPU

Figure 7: Attacks against passive external memory

In Figure 7, the grey box represents the SoC with the 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 41. There are no additional Organisational Security Policies defined in this package.

7.1.1.4 Assumption

Application Note 42. 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 43. 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 6: 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 authorised 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 Protection against an unauthorized rollback of memory content
1

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: FIA_UAU.1 or FDP_IRA.1

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 Irreversibility Anchor for external memory

1

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: FDP_IRA.1

There are no management activities foreseen.

Audit: FDP_IRA.1

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

- Minimal: 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 TOE shall implement an Irreversibility Anchor mechanism to verify the freshness of data stored in [assignment*: physically separated memory]*.

FDP_IRA.1.2 The Irreversibility Anchor shall provide a reference for [selection: *write, erase, [assignment: other operation that changes the content of the physically separated memory]*] transactions, such that that each transaction of this type shall be associated with a different value of the Irreversibility Anchor.

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 44. 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 passive external memory[28].

FDP_DAU.2.2/PM The TSF shall provide the 3S[29] 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[30] on behalf of the user to be performed before the user is identified.

FIA_UID.1.2/PM The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

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

[28] [assignment: *list of objects or information types*]

[29] [assignment: *list of subjects*]

[30] [assignment: *list of TSF-mediated actions*]

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

FPT_RPL.1.1/PM The TSF shall detect replay for the following entities: commands issued by the 3S to the passive external memory for the read, write and erase operations.[31]

FPT_RPL.1.2/PM The TSF shall perform [assignment: *list of specific actions*] when a replay is detected.

Application Note 45. The actions applied in case a replay is detected are considered to be product specific. Therefore, the assignment in FPT_RPL.1.2/PM needs to be completed by the author of the Security Target.

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: FIA_UAU.1 or FDP_IRA.1

FDP_URC.1.1/PM The TOE shall detect an unauthorised replacement of the content stored in passive external memory[32] 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[33].

FDP_IRA.1.2/PM The Irreversibility Anchor shall maintain a distinct transaction reference for each write, erase[34] operation and that is unambiguously linked with the current content of the transaction with the associated physically separated memory.

[31] [assignment: *list of identified entities*]

[32] [assignment: *physically separated memory*]

[33] [assignment: *physically separated memory*]

[34] [selection: *write, erase, [assignment: further operation that changes the content of the 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[35].
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**
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FDP_SDC.1.1/PM	The TSF shall ensure the confidentiality of the following user data: information of the user data[36] while it is stored in the persistent memory[37].

Application Note 46. Persistent memory in this iteration of FDP_SDC.1 refers to external memory and no to any other memory.

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*].
Refinement:	**This SFR applies for passive external memory.**

7.1.4.2 Rationale for the SFRs

Table 7 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

[35] [selection: *operation, power off, power saving, any operation mode*]

[36] [selection: *all user data, the following user data [assignment: list of user data*]]

[37] [selection: *temporary memory, persistent memory, any memory*]

Objective	TOE Security Functional and Assurance Requirements
O.Pas-Mem-Unauth-Rollback-Prot	FDP_URC.1/PM for Protection against an unauthorised rollback of content Supported by: FDP_IRA.1/PM for Irreversibility Anchor of external memory 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 7: 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:

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 SFR FDP_IRA.1/PM unambiguously links the current content of the transaction with the associated physically separated memory to a distinct transaction references and thereby ensures that an unauthorised replacement of the memory content is detected.

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	FIA_UAU.1 or FDP_IRA	Satisfied by FDP_IRA.1/PM
FDP_DAU.2/PM	FIA_UID.1	Satisfied by FIA_UID.1/PM

Table 8: 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 47. 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.

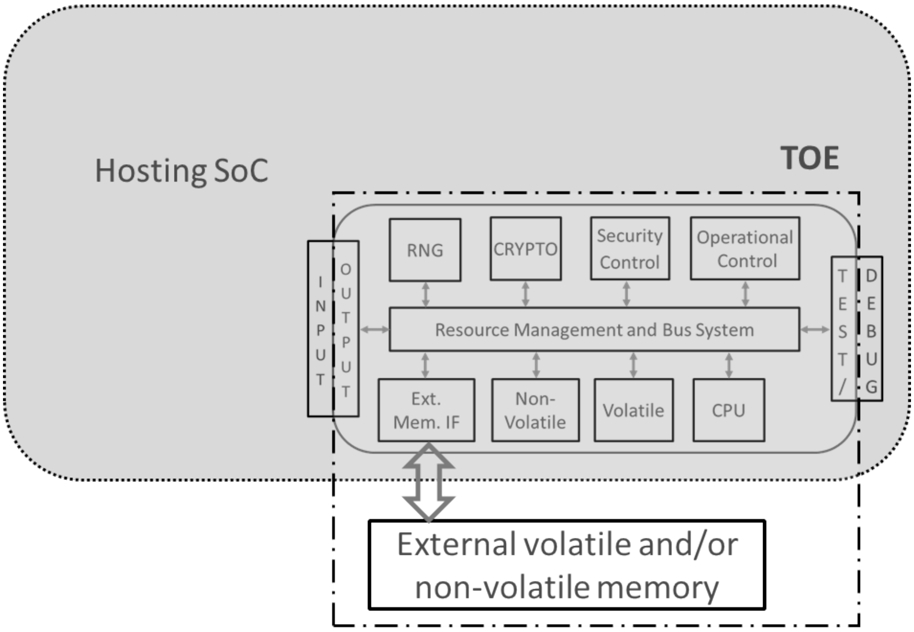


Figure 8: 3S with secure external memory

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 48. 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.

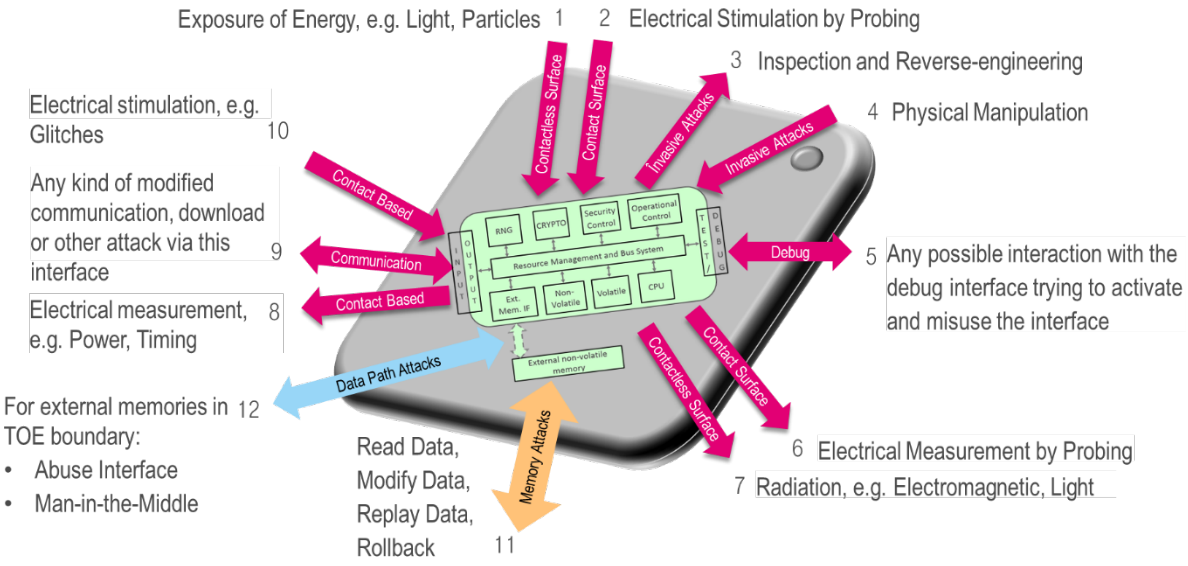

Exposure of Energy, e.g. Light, Particles 1
2 Electrical Stimulation by Probing
3 Inspection and Reverse-engineering
4 Physical Manipulation
Electrical stimulation, e.g. Glitches 10
Any kind of modified communication, download or other attack via this interface 9
Electrical measurement, e.g. Power, Timing 8
5 Any possible interaction with the debug interface trying to activate and misuse the interface
For external memories in TOE boundary: 12
• Abuse Interface
• Man-in-the-Middle
Read Data, Modify Data, Replay Data, Rollback 11
6 Electrical Measurement by Probing
7 Radiation, e.g. Electromagnetic, Light
Contactless Surface
Contact Surface
Invasive Attacks
Invasive Attacks
Contact Based
Communication
Contact Based
Debug
Data Path Attacks
Memory Attacks
Contactless Surface
Contact Surface
INPUT
OUTPUT
RNG
CRYPTO
Security Control
Operational Control
Resource Management and Bus System
Ext. Mem. IF
Non-Volatile
Volatile
CPU
TEST/
DEBUG
External non-volatile 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 49. 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 50. 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 9: 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 assumption "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 51. 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, see section 7.1.3.

7.2.3.1 Definition of the Family FDP_URC

Application Note 52. 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 53. 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 secure external memory[38].
FDP_DAU.2.2/SM	The TSF shall provide the 3S[39] 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 secure external 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.
FIA_UID.1.1/SM	The TSF shall allow the secure start-up or wake-up without access to user data[40] 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 secure external memory is required before further actions are performed. Based on the unambiguous assignment only one instance of the secure external 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

[38] [assignment: *list of objects or information types*]

[39] [assignment: *list of subjects*]

[40] [assignment: *list of TSF-mediated actions*]

Dependencies: No dependencies

FPT_RPL.1.1/SM The TSF shall detect replay for the following entities: commands issued by the 3S to the secure external memory for the read, write and erase operations[41].

FPT_RPL.1.2/SM The TSF shall perform [assignment: *list of specific actions*] when a replay is detected.

Application Note 54. The actions applied in case a replay is detected are considered to be product specific. Therefore, the assignment in FPT_RPL.1.2/SM needs to be completed by the author of the Security Target.

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: FIA_UAU.1 or FDP_IRA.

FDP_URC.1.1/SM The TOE shall detect an unauthorised replacement of the contents stored in secure external memory[42] 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.

FDP_IRA.1.1/SM The TSF shall verify the freshness of data for each read operation from the secure external memory[43].

FDP_IRA.1.2/SM The Irreversibility Anchor shall maintain a distinct transaction references for each write, erase[44] 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[45].

[41] [assignment: *list of identified entities*].

[42] [assignment: *physically separated memory*]

[43] [assignment: *physically separated memory*]

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

[45] [selection: *operation, power off, power saving, any operation mode*]

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[46].

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*].

Refinement: **This SFR applies for secure external memory.**

The security functional requirements “Basic internal transfer protection (FDP_ITT.1)” and “Basic internal TSF data transfer protection (FPT_ITT.1)” have been selected to ensure that the secure external memory as part of the TOE must resist leakage attacks (both for user data and TSF data). The corresponding security policy is defined in the security functional requirement “Subset information flow control (FDP_IFC.1)”.

The TOE shall meet the requirement “Basic internal transfer protection (FDP_ITT.1/SM)” as specified below.

FDP_ITT.1/SM **Basic internal transfer protection**

Hierarchical to: No other components.

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

FDP_ITT.1.1/SM The TSF shall enforce the Storage Processing Policy[47] to prevent the disclosure[48] of user data when it is transmitted between physically-separated parts of the TOE.

Refinement: **The memory matrix, the controller or CPU integrated in the secure external memory and other functional units of the secure external memory (as part of the TOE) are seen as physically-separated parts of the TOE.**

[46] [assignment: *memory area*]

[47] [assignment: *access control SFP(s) and/or information flow control SFP(s)*]

[48] [selection: *disclosure, modification, loss of use*]

The TOE shall meet the requirement “Basic internal TSF data transfer protection (FPT_ITT.1/3S)” as specified below.

FPT_ITT.1/SM **Basic internal TSF data transfer protection**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_ITT.1.1/SM The TSF shall protect TSF data from disclosure[49] when it is transmitted between separate parts of the TOE.

Refinement: **The memory matrix, the controller or CPU integrated in the secure external memory and other functional units of the secure external memory (as part of the TOE) are seen as physically-separated parts of the TOE.**

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

The TOE shall meet the requirement “Subset information flow control (FDP_IFC.1/3S)” as specified below:

FDP_IFC.1/SM **Subset information flow control**

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1/3S The TSF shall enforce the Storage Processing Policy[50] on all confidential data when they are processed or transferred by the TOE[51].

The following Security Function Policy (SFP) Storage Processing Policy is defined for the requirement “Subset information flow control (FDP_IFC.1/SM)”:

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

Application Note 55. The three SFR FDP_ITT.1, FDP_ITT.1 and FDP_IFC.1 together with the refinement define the protection of data in transit against leakage. This covers the transfer between 3S and secure external memory as well as the processing of data inside the secure external memory (the processing of date in the 3S is covered in the base PP). This comprises confidentiality of the TSF data as well as confidentiality of user data.

7.2.4.2 Rationale for the SFRs

Table 10 below provides an overview, how the SFRs are combined to meet the security objectives. The justification for each objective is detailed after the table.

[49] [selection: *disclosure, modification*]

[50] [assignment: *information flow control SFP*]

[51] [assignment: *list of subjects, information, and operations that cause controlled information to flow to and from controlled subjects covered by the SFP*]

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 Supported by: FDP_IRA.1/SM for Irreversibility Anchor of external 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	FDP_ITT.1/SM Basic internal transfer protection FPT_ITT.1/SM Basic internal TSF data transfer protection FDP_IFC.1/SM Subset information flow control

Table 10: 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 SFR FDP_IRA.1/SM unambiguously links the current content of the transaction with the associated physically separated memory to a distinct transaction references and thereby ensures that an unauthorised replacement of the memory content is detected.

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 the 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:

The SFRs FPT_ITT.1/SM and FDP_ITT.1/SM together with the policy statement in FDP_IFC.1/SM require the TOE to prevent leakage of TSF data and user data when transferred between the 3S and the secure external memory and during processing in the secure external memory. Therefore, these SFRs address 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	FIA_UAU.1 or FDP_IRA	Satisfied by FDP_IRA.1/SM
FDP_DAU.2/SM	FIA_UID.1	Satisfied by FIA_UID.1/SM
FDP_ITT.1/SM	**FDP_ACC.1 or FDP_IFC.1**	**Yes by FDP_IFC.1/SM**
FDP_IFC.1/SM	**FDP_IFF.1**	**See discussion below**
FPT_ITT.1/SM	**None**	**No dependency**

Table 11: Overview of Dependencies of the SFRs for secure external memory

Part 2 of the Common Criteria defines the dependency of FDP_IFC.1 (information flow control policy statement) on FDP_IFF.1 (Simple security attributes). The specification of FDP_IFF.1 would not capture the nature of the security functional requirement nor add any detail. As stated in the Data Processing Policy referred to in FDP_IFC.1/SM there are no attributes necessary. The security functional requirement for the TOE is sufficiently described using FDP_ITT.1/SM and its Storage Processing Policy (FDP_IFC.1/SM).

7.3 Package for Loader Functionality

7.3.1 Security Problem Definition

7.3.1.1 Description of Assets

Application Note 56. There are no additional assets defined in this package.

7.3.1.2 Threat

Application Note 57. 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 58. 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

	O.Ctrl-Auth-Loader	OE.Loader-Usage
P.Access-Ctlr-Loader	X	X

Table 12: 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 59. 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

(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 60. 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 - Load (FDP_ACF.1/Load)” is specified as follows.

FDP_ACF.1/Load **Security attribute based access control - Load**

Hierarchical to: No other components.

Dependencies: FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/Load The TSF shall enforce the Loader SFP[52] to objects, based on the following:

(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*][53].

FDP_ACF.1.2/Load 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 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 61. 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/Load	Inter-TSF trusted channel
	FDP_UCT.1/Load	Basic data exchange confidentiality
	FDP_UIT.1/Load	Data exchange integrity
	FDP_ACC.1/Load	Subset access control – Load
	FDP_ACF.1/Load	Security attribute based access control - Load

Table 13: Mapping between Objectives and SFRs for the Loader

[52] [assignment: *access control SFP*]

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

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/Load require the TSF to implement access control for the Loader functionality.

The SFR FTP_ITC.1/Load, FDP_UCT.1/Load and FDP_UIT.1/Load 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.

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	FMT_MSA.1 FMT_SMR.1	The dependencies FMT_MSA.3 is not satisfied, see the rationale below the table

Table 14: Overview of SFR dependencies for the Loader package

The SFR FMT_MSA.3 and its dependencies 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 Package for Cryptographic Services

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

Application Note 62. 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 63. 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.

Application Note 64. The term "purpose" shall be replaced by the cryptographic algorithm that are implemented by the TOE. The objective O.Crypto-Service shall be integrated for each cryptographic algorithm to support the mapping to the associated Security Functional Requirements.

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 15: 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, or FCS_CKM.5 Cryptographic key derivation] FCS_CKM.3 Cryptographic key access
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 65. 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 66. The cryptographic operations defined in [11] 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.6/iteration" as specified below.

FCS_CKM.6/iteration	**Timing and event of 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.6.1/iteration	The TSF shall destroy [assignment: *list of cryptographic keys (including keying material)*] when [selection: *no longer needed, [assignment: other circumstances for key or keying material destruction]*].
FCS_CKM.6.2/iteration	The TSF shall destroy cryptographic keys and keying material specified by FCS_CKM.6.1 in accordance with a specified cryptographic key destruction method [assignment: *cryptographic key destruction method*] that meets the following: [assignment: *list of standards*].

Application Note 67. The ST author shall provide iterations of FCS_CKM.6 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.6 can meet the dependency for various cryptographic algorithms defined with FCS_COP1.

7.4.4.2 Rationale for the SFRs

The FCS_COP.1/iteration and FCS_CKM.6/iteration 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, or FCS_CKM.5 FCS_CKM.3	
FCS_CKM.6/iteration	FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1	

Table 16: Overview of SFR dependencies for the Loader package

The dependency of FCS_COP.1 on FCS_CKM.6 is fulfilled within the package

FCS_COP.1 has a dependency with [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, or FCS_CKM.5 Cryptographic key derivation]. This PP leaves 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.

In case of FCS_CKM.6, the SFR has a similar dependency to the previously commented for FCS_COP.1, excluding FCS_CKM.5 of the dependency. Therefore, the dependency is not satisfied and the ST author shall include the respective SFR component according to method used as source for the keys.

In addition, FCS_COP.1 has a dependency with FCS_CKM.3. This PP leaves the decision to the ST author to include the SFR if the TOE implements it or provide a justification for the missing dependency in case of not implementing the SFR.

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

7.5.1.2 Threats

Application Note 69. 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 70. 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[54] 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.

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

7.5.2.2 Security Objectives for the TOE Environment

Application Note 71. This package does not include additional Security Objectives for the TOE Environment.

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 17: 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/SW_TSF	**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/SW_TSF	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[55].
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 72. 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*].
Refinement:	**This SFR FMT_SMF.1/SW_TSF defines the management function provided by the 3S to FW and SW for the processing of keys and attributes enforcing the protection and use of TSF data.**

Application Note 73. 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 74. 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.

[55] [assignment: *the authorised identified roles*]

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 75. 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 76. 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*].
Refinement:	**The SFR FMT_SMF.1/SWIso defines the management function provided by the 3S to enforce the access control policy to memories and resources.**

7.5.4.2 Rationale for the SFRs

Table 18 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/SW_TSF 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 18: Mapping between Objectives and SFRs for the Software Isolation Package

The SFR FMT_MTD.1/SW_TSF 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/SW_TSF 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/SW_TSF	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.2/SWIso (because it is hierarchical to FDP_ACC.1) FMT_SMF.1/SWIso not satisfied, see the rationale below the table
FMT_SMF.1/SWIso	No dependency	

Table 19: 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.

7.6 Package for Secure Update

This optional package proposes an extension to allow the secure update of the TOE.

An updatable system is crucial in maintaining the security of a 3S over time. Software or firmware of 3S out in the field could have been released with unnoticed vulnerabilities. On the other hand, attackers are constantly evolving their tactics, and new threats emerge regularly, making it critical to have a way to quickly update software to address vulnerabilities and reduce the attack surface. Leaving a vulnerable software or firmware unpatched would severely increase the risk of exploitation.

In addition to security concerns, there are also functional reasons for having an update mechanism. As new features and improvements are developed, an update mechanism allows for these changes to be easily deployed to end-users. This ensures that the product remains current and competitive in the marketplace, and it also provides a way to fix bugs that impact functionality.

Therefore, it is essential to ensure that any update mechanism included in the product is well-protected and well-functioning. By supporting this feature, it is possible to update the TOE firmware and/or software in order to replace the logic producing the vulnerability by new code where the vulnerability is removed. The update mechanism shall also include rollback protection, preventing the installation of TOE code that is not newer than that in the initial TOE, except in optional authorized and intended situations.

The following terminology related to secure update (also called patch management) is used in this package:

- **Update:** type of binary code intended to introduce additions or modifications of a functional or security feature. In the context of this package, this term can refer to either an update that targets a specific area of the code (which is usually known as a *Patch),* or it may also refer to a entire replacement of the full firmware or software update of the TOE.
- **Initial TOE**: TOE that supports evaluated features allowing at least to securely load and install update(s), without any applied *updates*.
- **Final TOE:** TOE after applying the *updates*.
- **Activation:** operation performed on a *update* to transform the *initial TOE* into the *final TOE*.
- **Loader:** piece of the TSF of the *initial TOE* that implements the loading and *activation* of an *update*
- **Update verification**: technical mechanism to verify the integrity and authenticity of an update.
- **End of support**: date until when the user may expect to receive new *updates*.
- **Identification data:** data that identifies the *initial TOE*, the applied *update(s)* or the final *TOE*.

This package also mentions the following roles of users involved in update management activities:

- **Update Issuer:** user(s) responsible for generation, protection and distribution of updates to end users of the TOE. Normally, this role belongs to the same organization as the TOE manufacturer.
- **Update Deployer:** user(s) in the operational environment of the TOE that are responsible for TOE updating tasks, such as checking for new updates and installation or scheduling of updates. They are also responsible for providing the necessary means to make the updates distributed by the *Update Issuer* available to the *Loader* for their activation.

The updated code is provided in the form of security updates (or simply updates), provided in binary form. This package defines a series of additional functional requirements for the TOE to be able to apply these updates in order to update its code. As previously mentioned, as a result of applying an update, the Initial TOE becomes what is called the Final TOE. The application of an update is carried out by a component of the TSF which is conceptually named as Loader. The loader performs various operations in order to apply the update, including loading (reading update contents from a location), activation (patching or replacing existing code with the code of the update). Moreover, the loader shall be responsible for update verification before applying it.

The scope of the TSF in this package doesn't include functionality for the TOE to check the existence of new updates; this functionality belongs to the Update Deployer. However, a TOE compliant with this package shall provide mechanisms for the Update Deployer to obtain updates through a secure channel and to force the TOE to activate them.

The solution provided in this package relies on additional functional requirements (FPT_UPM) which address the patch or update functionality of the initial TOE.

7.6.1 Security Problem Definition

7.6.1.1 Description of Assets

Application Note 77. There are no additional assets defined in this package.

7.6.1.2 Threats

The TOE shall avert the threat "Discovery and exploitation of vulnerabilities in the TOE (T.Vulnerability)" as specified below.

T.Vulnerability

Discovery and exploitation of vulnerabilities in the TOE

An attacker may find and exploit a vulnerability in the TOE (e.g., a bug or a protection of a security asset that becomes outdated) for a TOE out in the field, in order to gain unauthorized access to the TOE assets, cause harm to the TOE, or impact the security of the TOE.

The TOE shall avert the threat "Blocking of update mechanism (T.Update-Blocking)" as specified below.

T.Update-Blocking

Blocking of update mechanism

An attacker is able to block the ability of the TOE to get new security updates, so the TOE is not able to receive a security update, remaining in a state where future detected security flaws will not be corrected.

The TOE shall avert the threat "Forging of malicious updates (T.Update-Forging)" as specified below.

T.Update-Forging

Forging of malicious updates

An attacker forges a rogue malicious update that is installed or processed by the TOE, altering the intended TSF functionality.

The TOE shall avert the threat "Eavesdrop during update transport (T.Update-TransportEavesdrop)" as specified below.

T.Update-TransportEavesdrop

Eavesdrop during update transport

An attacker eavesdrops on the communication channel using to transport the updates between the Update Issuer and the TOE, enabling him to access to the data being transported without authorization.

The TOE shall avert the threat "Rollback through updating (T.Update-Rollback)" as specified below.

T.Update-Rollback

Rollback through updating

An attacker uses the update mechanism to install an update containing a version of the TOE code that is not newer than that installed in the initial TOE before the moment of update activation, out of an authorized scenario. This could potentially allow the attacker to replace secure TOE code with older and vulnerable code.

7.6.1.3 Organisational Security Policy

The TOE shall implement the policy "Regular checks for updates (P.Update-RegularChecks)" as specified below.

P.Update-RegularChecks

Regular checks for updates

Update Deployers, responsible for updating of the TOE, regularly check for new updates.

7.6.1.4 Assumptions

A.Update-Management

Update management by Update Deployers

Update Deployers take required measures to allow reception of update notifications, loading and activation of the updates, in order to support any activity which is required to perform the updating process, such as availability of the direct or indirect communication channels required to obtain the update and making it available to the loader. They also verify that the TOE has correctly received and activated the update.

7.6.2 Security Objectives

7.6.2.1 Security Objectives for the TOE

The TOE shall provide the "TOE Code Update Mechanism (O.Code-Update)" as specified below.

O.Code-Update

TOE Code Update Mechanism

The TOE shall implement a software update mechanism that allows the Update Deployer to update parts of the TOE software or firmware.

The TOE shall provide “Secure communication channel for update retrieval (O.Update-SecureTransport)” as specified below.

O.Update-SecureTransport Secure communication channel for update retrieval

The TOE shall establish a secure communication channel for retrieval of updates that prevents unauthorized access to the contents being transported.

The TOE shall provide “Authenticated update installation (O.Update-AuthenticatedInstall)” as specified below.

O.Update-AuthenticatedInstall Authenticated update installation

An administrator user with required update privileges shall be required to install an update or schedule update installation.

The TOE shall provide “Secure update load (O.Secure-UpdateLoad)” as specified below.

O.Secure-UpdateLoad Secure update load

The Loader of the Initial TOE shall check the authenticity and integrity of the loaded update. The TOE shall allow updating the critical security parameters used to verify the authenticity and integrity of the loaded update.

The TOE shall provide “Atomic update activation and update of identification data (O.Atomic-UpdateActivation)” as specified below:

O.Atomic-UpdateActivation Atomic update activation and update of identification data

Activation of the update and update of the Identification Data shall be performed in an atomic way. All the operations needed for the code to be able to operate as in the Final TOE shall be completed before activation. If the Atomic Activation is successful, then the resulting product is the Final TOE.

The TOE shall provide “Anti-rollback during updating (O.Update-AntiRollback)” as specified below:

O.Update-AntiRollback Anti-rollback during updating

Activation of an update shall be blocked if the version of the TOE code in the update is not newer than the version of the code subject of updating in the initial TOE, except in determined authorized and optional situations.

The TOE shall provide “Secure update failure (O.Secure-UpdateFailure)” as specified below:

O.Secure-UpdateFailure Secure update failure

In case of interruption or incident which prevents the forming of the Final TOE (such as tearing, integrity violation, error case…), the Initial TOE shall remain in its initial state or fail secure.

7.6.2.2 Security Objectives for the TOE Environment

The operational environment of the TOE shall provide “Update management by Update Deployers (OE.Update- Management)” as specified below.

OE.Update-Management — Update management by Update Deployers

Update deployers shall take required measures to allow reception of update notifications, loading, installation and activation of the update, in order to support any activity which is required to perform the updating process, such as availability of the direct or indirect communication channels required to obtain the update and making it available to the loader. They also shall verify that the TOE has correctly received and activated the update.

The operational environment of the TOE shall provide “Regular checks of new updates (OE.Update-RegularChecks)” as specified below.

OE.Update-RegularChecks — Regular checks of new updates

Update Deployers, responsible for updating of the TOE, shall regularly check for new updates.

The operational environment of the TOE shall provide “Update availability (OE.Update-Availability)” as specified below.

OE.Update-Availability — Update Availability

The Update Issuer shall notify the Update Deployer of the availability of new updates, and shall make available in a secure way, security updates and installation instructions until the end of support of the TOE.

7.6.2.3 Security Objectives Rationale

	O.Code-Update	O.Update-AuthenticatedInstall	O.Secure-UpdateLoad	O.Atomic-UpdateActivation	O.Update-AntiRollback	O.Secure-UpdateFailure	O.Update-SecureTransport	OE.Update-Management	OE.Update- RegularChecks	OE.Update-Availability
T.Vulnerability	X									
T.Update-Blocking								X	X	X
T.Update-Forging		X	X	X		X				

T.Update-TransportEavesdrop							X			
T.Update-Rollback					X					
P.Update-RegularChecks									X	
A.Update-Management								X		

Table 20 Mapping from Security Problem Definition to objectives when the environment is responsible for update checking

The threat “Discovery and exploitation of vulnerabilities in the TOE (T.Vulnerability)”is mitigated by the security objective for the TOE “TOE Code Update Mechanism (O.Code-Update)”, which ensures that the TOE implements a mechanism that allows to update the TOE software or firmware, enabling it to replace vulnerable code with new code that is not vulnerable.

The threat “Blocking of update mechanism (T.Update-Blocking)” is mitigated by a joint effort of the TOE and the operational environment as follows:

- “Update management by Update Deployers (OE.Update-Management)” ensures that Update Deployers will take the required measures to ensure that they are aware of the release of new updates and that such updates are adequately loaded and activated.
- “Regular checks of new updates (OE.Update-RegularChecks)” ensures that Update Deployers responsible for updating, regularly check the existence of new updates.
- “Update availability (OE.Update-Availability)” ensures that the Update Issuer notifies the Update Deployer about the release of new updates, and that these are made available to users in a secure manner, along with installation instructions.

The threat “Forging of malicious updates (T.Update-Forging)” is mitigated as a joint effort of the TOE and the operational environment as follows:

- “Authenticated update installation (O.Update-AuthenticatedInstall)” ensures that every update installation is approved by an administrative entity.
- “Secure update load (O.Secure-UpdateLoad)” ensures that the TOE itself has mechanisms to verify the signature of the update.
- “Atomic update activation and update of identification data (O.Atomic-UpdateActivation)” ensures that, only after successful verification of the signature, the TOE will process and install the update in an atomic way, so no dangerous TSF mediated actions are allowed.
- “Secure update failure (O.Secure-UpdateFailure)” provides that the TOE will prevent the operation of the TOE in a failure state, restoring the TOE to its initial state.

The threat “Eavesdrop during update transport (T.Update-TransportEavesdrop)” is mitigated by the TOE as follows: the security objective for the TOE “Secure communication channel for update retrieval (O.Update-SecureTransport)” ensures that the TOE is capable to establish a secure communication channel to retrieve updates issued by the Update Issuer, preventing unauthorized access to the contents being transmitted.

The threat “Rollback through updating (T.Update-Rollback)” is mitigated by the TOE as follows: the security objective for the TOE “Anti-rollback during updating (O.Update-AntiRollback)” ensures that updating operations are blocked unless the TOE code in the update is newer than the code in the TOE at the moment before activating the update, except in authorized cases.

The organisational security policy “Regular checks for updates (P.Update-RegularChecks)” is directly enforced by the objective for the operational environment “Regular checks of new updates (OE.Update-RegularChecks)”, which ensures that Update Deployers, responsible for updating, regularly check the existence of new updates.

The assumption “Update management by Update Deployers (A.Update-Management)” is directly upheld by the objective for the operational environment “Update management by Update Deployers (OE.Update-Management)”, which ensures that Update Deployers, responsible for updating, will take the required measures to ensure that they are aware of the release of new updates and that such updates are adequately loaded and activated, and also that they provide the necessary means to make available the direct or indirect communication channel used to obtain the updates.

7.6.3 Extended Component Definition

This package defines the extended security functional requirement family “Protection of the TSF during update management (FPT_UPM)”.

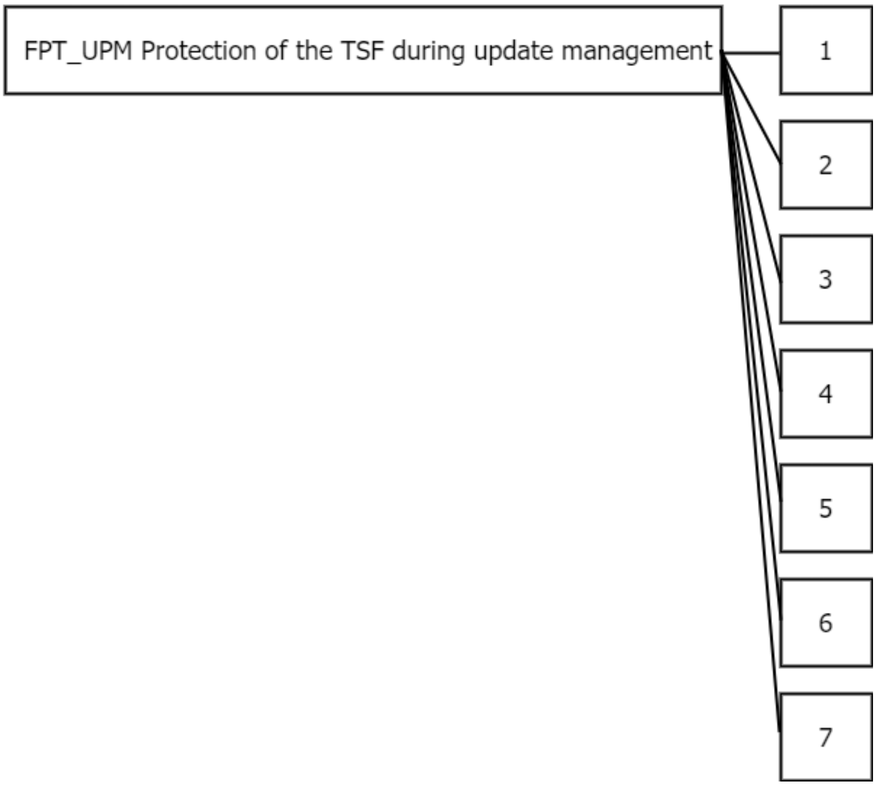
Update Management is a whole new topic that was not previously contemplated in Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components [2]. A new family is needed to describe most of the features expected to be present in an updatable TOE. The family is included under FPT_UPM as it deals with the protection of the TSF from malicious updates and from an unnoticed unsecure state due to the presence of vulnerabilities not detected during the initial evaluation.

7.6.3.1 Definition of the family FPT_UPM

Family behavior

This family covers the requirements needed to provide an update mechanism, and to protect the TSF while applying new updates to the Initial TOE.

Component levelling



FPT_UPM.1 provides a code update mechanism that allows to update the TOE software or firmware through update activation.

FPT_UPM.2 provides the capability to verify that an administrative user or an administration terminal are authenticated in order to allow application or scheduling of updates.

FPT_UPM.3 provides verification cryptographic signature of updates, with a minimum strength and meeting a list of standards, before being able to install them.

FPT_UPM.4 provides that update activation is performed in an atomic way, limiting the TSF mediated actions that are allowed.

FPT_UPM.5 provides protection against code downgrading during update process.

FPT_UPM.6 provides a secure state when failures occur during update process.

FPT_UPM.7 provides a secure communication channel for update retrieval.

Management FPT_UPM.1, FPT_UPM.2, FPT_UPM.3, FPT_UPM.4, FPT_UPM.6, FPT_UPM.7

There are no management activities foreseen.

Audit FPT_UPM.1, FPT_UPM.2 FPT_UPM.3, FPT_UPM.4, FPT_UPM.6, FPT_UPM.7

There are no actions defined to be auditable.

FPT_UPM.1	**Code Updating**
Hierarchical to:	No other components.
Dependencies:	No dependencies

FPT_UPM.1.1 The TSF shall provide an update mechanism that allows updating [assignment: *list of the parts of the TOE software or firmware subject to the mechanism*] through update activation.

FPT_UPM.2 Administrator mediated updating

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating

FPT_UPM.2.1 The TSF shall require an authenticated [selection: *administrative user with update privileges, administration terminal with update privileges*] to allow the [selection: *application, scheduling*] of updates.

FPT_UPM.3 Trusted updating

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating

FPT_UPM.3.1 The TSF shall cryptographically verify updates prior to installation using a digital signature scheme that provides a strength of [assignment: *positive integer*] bits that meet the following [assignment: *list of standards*], blocking update installation if the verification fails.

FPT_UPM.3.2 The TSF shall allow the secure update of the critical security parameters involved in the verification of the digital signature.

FPT_UPM.4 Atomic update activation

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating, FPT_UPM.3 Trusted updating

FPT_UPM.4.1 The TSF shall perform the activation of the update in an atomic way so that it will not perform any TSF mediated action but [assignment: allowed actions performed by the TSF].

FPT_UPM.4.2 After atomic update activation the TOE shall show the new version.

FPT_UPM.5 Rollback protection in update activation

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating

FPT_UPM.5.1 The TSF shall block the activation of an update, if the version of the code in the update is not newer than that of the code to be replaced or patched in the initial TOE, except in the following situations: [assignment: *situations in which applying an update with a version not newer than the code in the initial TOE is allowed*].

FPT_UPM.6 Preservation of secure state during updating

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating, FPT_UPM.3 Trusted updating, FPT_UPM.4 Atomic update activation, FPT_UPM.5 Rollback protection in update activation.

FPT_UPM.6.1 The TSF shall preserve a secure state when the following types of failures occur during updating:

a) Failure according to FPT_UPM.3 Trusted updating.
b) Failure according to FPT_UPM.4 Atomic update activation.
c) Failure according to FPT_UPM.5 Rollback protection in update activation

FPT_UPM.6.2 The following rules and actions will be performed when a failure is detected. [assignment: *rules describing actions and conditions regarding failures*].

FPT_UPM.7 **Secure communication channel for update retrieval**

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating

FPT_UPM.7.1 The TSF shall provide a communication channel between itself and [selection, choose one of: *the Update Issuer's update server, [assignment: other update server*]] that provides authentication of the server and protection of the channel data from [selection, choose one of: *modification, modification and disclosure*].

FPT_UPM.7.2 The TSF shall permit [selection, choose one of: *the TSF, the update server*] to initiate communication via the trusted channel.

7.6.4 IT Security Requirements

7.6.4.1 SFRs for the TOE

FPT_UPM.1 **Code Updating**

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_UPM.1.1 The TSF shall provide an update mechanism that allows updating [assignment: *list of the parts of the TOE software or firmware subject to the mechanism*] through update activation.

FPT_UPM.2 **Administrator mediated updating**

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating

FPT_UPM.2.1 The TSF shall require an authenticated [selection: *administrative user with update privileges, administration terminal with update privileges*] to allow the [selection: *application, scheduling*] of updates.

FPT_UPM.3 **Trusted updating**

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating

FPT_UPM.3.1 The TSF shall cryptographically verify updates prior to installation using a digital signature scheme that provides a strength of [assignment: *positive integer*] bits that meet the following [assignment: *list of standards*], blocking update installation if the verification fails.

FPT_UPM.3.2 The TSF shall allow the secure update of the critical security parameters involved in the verification of the digital signature.

FPT_UPM.4 Atomic update activation

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating, FPT_UPM.3 Trusted updating,

FPT_UPM.4.1 The TSF shall perform the activation of the update in an atomic way so that it will not perform any TSF mediated action but [assignment: *allowed actions performed by the TSF*].

FPT_UPM.4.2 After atomic update activation the TOE shall show the new version.

FPT_UPM.5 Rollback protection in update activation

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating

FPT_UPM.5.1 The TSF shall block the activation of an update, if the version of the code in the update is not newer than that of the code to be replaced or patched in the initial TOE, except in the following situations: [assignment: *situations in which applying an update with a version not newer than the code in the initial TOE is allowed*].

Application Note 78. If an update rollback protection mechanism permits downgrading to an older version of the TOE code or re-installing the existing version, the ST author shall specify all situations in which these operations are allowed in the FPT_UPM.5 assignment. On the other hand, if the TOE does not allow the installation of an update with a code version that is not newer than the one in the initial TOE, the ST author should clearly indicate this in the assignment, e.g., by filling it in as "none”.

FPT_UPM.6 Preservation of secure state during updating

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating, FPT_UPM.3 Trusted updating, FPT_UPM.4 Atomic update activation, FPT_UPM.5 Rollback protection in update activation.

FPT_UPM.6.1 The TSF shall preserve a secure state when the following types of failures occur during updating:

a) Failure according to FPT_UPM.3 Trusted updating.
d) Failure according to FPT_UPM.4 Atomic update activation.
e) Failure according to FPT_UPM.5 Rollback protection in update activation

FPT_UPM.6.2 The following rules and actions will be performed when a failure is detected. [assignment: *rules describing actions and conditions regarding failures*].

FPT_UPM.7 **Secure communication channel for update retrieval**

Hierarchical to: No other components.

Dependencies: FPT_UPM.1 Code Updating

FPT_UPM.7.1 The TSF shall provide a communication channel between itself and [selection, choose one of: *the Update Issuer's update server, [assignment: other update server]*] that provides authentication of the server and protection of the channel data from [selection, choose one of: *modification, modification and disclosure*]

FPT_UPM.7.2 The TSF shall permit [selection, choose one of: *the TSF, the update server*] to initiate communication via the trusted channel.

Application Note 79. FPT_UPM.7.1 is meant to protect the communication channel used to transport updates between the Update Issuer and the TOE from unauthorized access. It includes at least protection of the data being transported in authenticity and integrity. Confidentiality is left as optional, with the second option of the second selection of FPT_UPM.7.1, and the ST author shall determine and indicate whether the read access to update contents on transit is also a target of the transport protection.

7.6.4.2 Rationale for the SFRs

Table 21 below describes 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.Code-Update	FPT_UPM.1
O.Update-AuthenticatedInstall	FPT_UPM.2
O.Secure-UpdateLoad	FPT_UPM.3
O.Atomic-UpdateActivation	FPT_UPM.4
O.Update-AntiRollback	FPT_UPM.5
O.Secure-UpdateFailure	FPT_UPM.6
O.Update-SecureTransport	FPT_UPM.7

Table 21: Mapping between Objectives and SFRs for the Package for Recovery of Stored User data

The justification related to the security objective “TOE Code Update Mechanism (O.Code-Update)” is as follows: FPT_UPM.1 requires that the TOE provides an update mechanism that allows updating the TOE software or firmware through update activation.

The justification related to the security objective “Authenticated update installation (O.Update-AuthenticatedInstall)” is as follows: FPT_UPM.2 requires that an authenticated administrative user or administration terminal with specific update permissions, in order to allow the application or scheduling of updates.

The justification related to the security objective “Secure update load (O.Secure-UpdateLoad)” is as follows: FPT_UPM.3 requires cryptographic verification of updates prior to installation using robust digital signatures, and allows the secure update of critical security parameters involved in the verification of the signature.

The justification related to the security objective “Atomic update activation and update of identification data (O.Atomic-UpdateActivation)” is as follows: FPT_UPM.4 requires that the activation of the update is carried out in an atomic way and that after activation the TOE shows the new version.

The justification related to the security objective “Anti-rollback during updating (O.Update-AntiRollback)” is as follows: FPT_UPM.5 requires that the activation of the update is blocked if the version of the TOE code in the update is not newer than the one installed in the TOE before updating, allowing defined exceptions for this restriction.

The justification related to the security objective “Secure update failure (O.Secure-UpdateFailure)” is as follows: FPT_UPM.6 requires the TSF to preserve a secure state upon failures during updating, and establishes rules and actions to perform when a failure is detected.

The justification related to the security objective “Secure communication channel for update retrieval (O.Update-SecureTransport)” is as follows: FPT_UPM.7 requires the TSF to establish a secure channel with the update server, ensuring authenticity, integrity and, optionally, confidentiality during transport, in order to retrieve new updates through this channel.

7.6.4.3 Dependencies of the SFRs

Requirement	Dependency	Satisfied Dependencies
FPT_UPM.1	N/A	N/A
FPT_UPM.2	FPT_UPM.1	FPT_UPM.1
FPT_UPM.3	FPT_UPM.1	FPT_UPM.1
FPT_UPM.4	FPT_UPM.1 FPT_UPM.3	FPT_UPM.1 FPT_UPM.3
FPT_UPM.5	FPT_UPM.1	FPT_UPM.1
FPT_UPM.6	FPT_UPM.1 FPT_UPM.3 FPT_UPM.4 FPT_UPM.5	FPT_UPM.1 FPT_UPM.3 FPT_UPM.4 FPT_UPM.5
FPT_UPM.7	FPT_UPM.1	FPT_UPM.1

Table 22 Dependencies for SFRs in Updater package

7.7 Package for Composite Software identity binding with asymmetric cryptography key

This optional package describes the key provisioning process that occurs during 3S in SoC Manufacturing or 3S in SoC Packaging (abbreviated as "3S in SoC manufacturing/packaging" throughout the text of this package), which provides asymmetric cryptography key material to the 3S in SoC before 3S in SoC delivery. The key material is subsequently used by the TOE to enable a Composite Software to cryptographically bind, or verify binding, of its identity to data it provides. The cryptographic binding might include, among other operations, attestation (i.e., cryptographically sign) or unwrapping (i.e., decryption) of data provided by a Composite Software.

The key material comprises of

- an asymmetric key pair generated, inside the TOE, or outside the TOE and securely injected into the TOE
- a certificate signing request containing the public key and issued outside of the TOE.

The asymmetric key pair is either unique per 3S instance (3S key pair) or shared among number of 3S instances (3S group key pair).

The term "provisioning" used, in this package, refers to the process of asymmetric key pair generation during 3S in SoC manufacturing/packaging, and certificate issuance. Likewise, the terms "provisioned private/public key" or "provisioned key pair", refers to the generated asymmetric key pair regardless of its generation method. The security problem described in this package is not dependent on the specific process used to generate the key pair, and is therefore agnostic to it.

The provisioning process takes place in a secure environment during 3S in SoC manufacturing/packaging and has the following flow:

- (Covered by a security objective for the environment) A keypair with a defined per-3S instance is generated inside the TOE, or outside the TOE and then injected. Uniqueness of the keypair per 3S instance or per group of 3S instances must be ensured.
- (Covered by a security objective for the environment) The public key of the key pair is embedded in a certificate signing request, and delivered to the certificate issuer outside of the TOE.
- (Covered by the environment) The certificate signing request is processed by the certificate issuer entity outside of the TOE, which issues the certificate by signing the request containing the public key.

Including this package in the security target means that the TOE will support key provisioning during the 3S in SoC manufacturing/packaging process.

By including this package, the TOE will have the capability, assumed to be available to any Composite Software in the field, to carry out cryptographic binding operations between data provided by Composite Software and its identity, using the provisioned private key. For example:

- Attestation enables a Composite Software to cryptographically prove its Composite Software identity combined with its 3S instance's identity, and transitively any additional data it provides, to entities outside the 3S. This use case is instantiated as a specific application of the general process outlined in Figure 10.

- Unwrapping enables a Composite Software to unwrap and decrypt data intended only for this Composite Software combined with its 3S identity, and coming from an entity outside the 3S, for which the TOE public key or certificate was provided. This use case is instantiated as a specific application of the general process outlined in Figure 11.

A TOE compliant with this package shall be able to perform cryptographic binding operations, on behalf of an authorized Composite Software, between input data and the identity of the Composite software. This binding operation is performed using the provisioned private key, and enables out-of-TOE entities to verify the cryptographic binding by using the public key included in the issued certificate. The process is summarized as follows, and depicted in the image below:

1. An authorized Composite Software provides to the TOE input data to be cryptographically bound with its identity.
2. The TOE verifies that the identified Composite SW is authorized to use the binding operation
3. The TOE uses the provisioned private key to cryptographically bind the provided data with the verified identity of the Composite SW.
4. The cryptographically bound data is returned to the calling Composite Software and provided to an out-of-3S identity.
5. The out-of-TOE entity uses the public key included in the issued certificate to verify the cryptographic binding in the data provided by the TOE, where successful verification indicates that the bound data originated in by the identified Composite Software in a genuine 3S hardware.

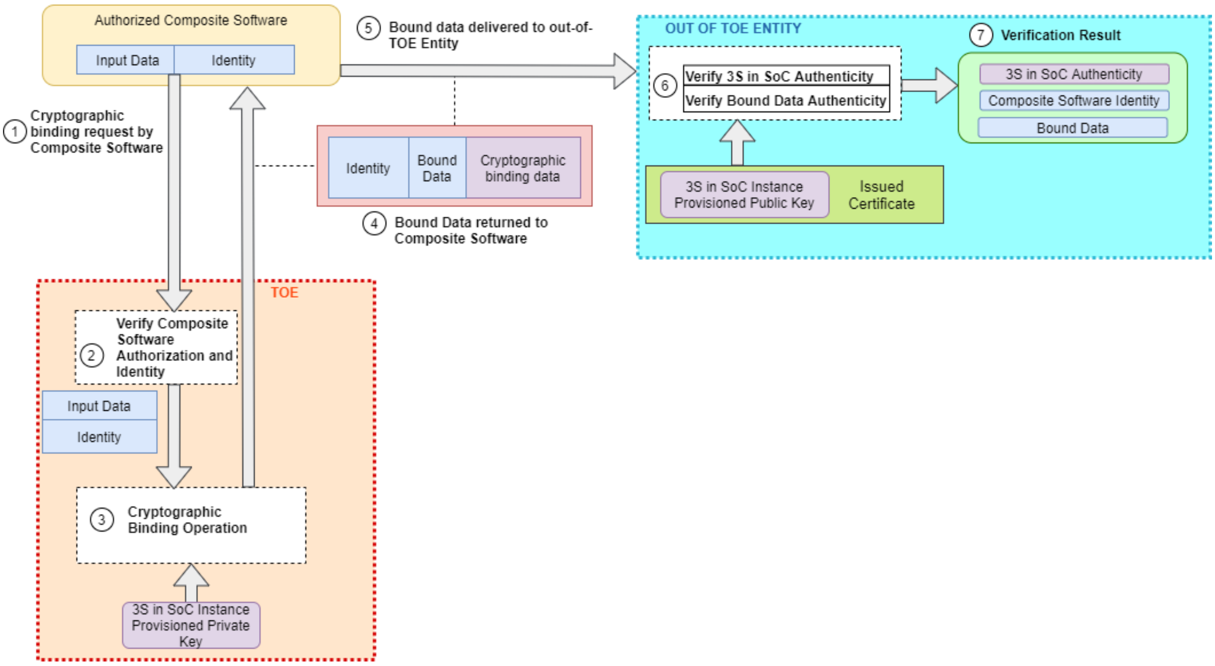

Authorized Composite Software
Input Data
Identity
5 Bound data delivered to out-of-TOE Entity
OUT OF TOE ENTITY
7 Verification Result
6
Verify 3S in SoC Authenticity
Verify Bound Data Authenticity
3S in SoC Authenticity
Composite Software Identity
Bound Data
1 Cryptographic binding request by Composite Software
Identity
Bound Data
Cryptographic binding data
4 Bound Data returned to Composite Software
3S in SoC Instance Provisioned Public Key
Issued Certificate
TOE
2 Verify Composite Software Authorization and Identity
Input Data
Identity
3 Cryptographic Binding Operation
3S in SoC Instance Provisioned Private Key

Figure 10 Cryptographic binding by the TOE and verification by an out-of-TOE entity

A TOE compliant with this package shall also be able to use the provisioned private key to cryptographically verify the binding performed by an out-of-TOE entity, between input data and the identity of the Composite Software that is intended to be recipient of the data. The process is summarized below and depicted in the following figure.

1. An out-of-TOE entity produces or receives input data to be cryptographically bound for a particular Composite Software identity.
2. The out-of-TOE entity uses the public key in the issued certificate to carry out a cryptographic binding operation between the data and the identity of the Composite Software that shall be recipient of such data.

3. The output of the cryptographic binding generated by the out-of-TOE identity is delivered to an authorized Composite Software.
4. The authorized Composite Software provides to the TOE the bound data.
5. The TOE verifies that the identified Composite SW is authorized to use the cryptographic binding verification operation.
6. The TOE uses the provisioned private key to verify the cryptographic binding between the provided data and the identity of the Composite Software, where a successful verification indicates that provided data was intended to the Composite Software on this 3S instance.
7. The TOE returns the result of the verification, and optionally data resulting from the cryptographic binding verification, to the authorized Composite Software.

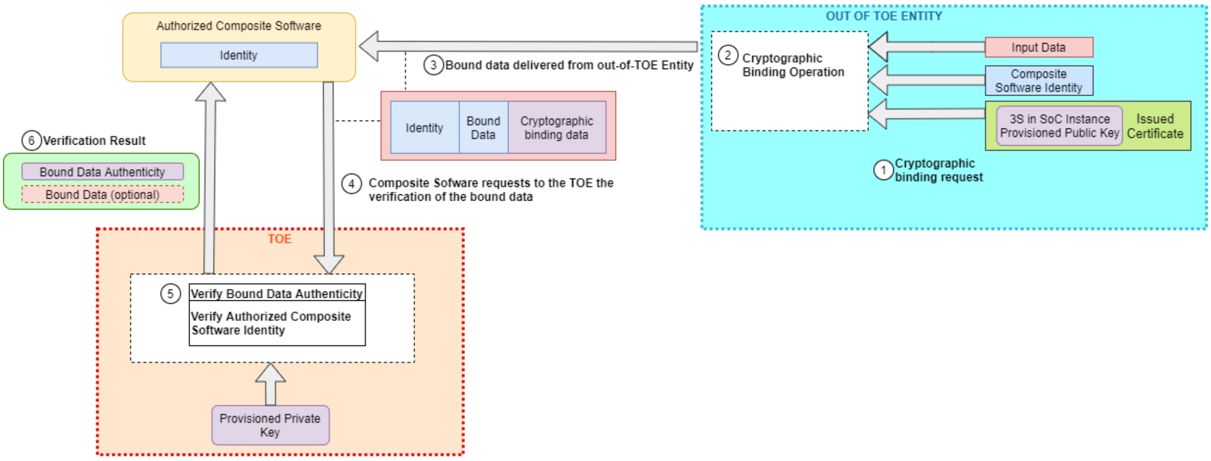

Authorized Composite Software
Identity
3 Bound data delivered from out-of-TOE Entity
OUT OF TOE ENTITY
2 Cryptographic Binding Operation
Input Data
Composite Software Identity
3S in SoC Instance Provisioned Public Key
Issued Certificate
1 Cryptographic binding request
Identity
Bound Data
Cryptographic binding data
6 Verification Result
Bound Data Authenticity
Bound Data (optional)
4 Composite Sofware requests to the TOE the verification of the bound data
TOE
5
Verify Bound Data Authenticity
Verify Authorized Composite Software Identity
Provisioned Private Key

Figure 11 Cryptographic binding by an out-of-TOE entity and verification by the TOE

Impersonation, in this package, is the ability of one Composite Software, to utilize the cryptographic capability offered by the TOE in this package, in order to (1) forge an output (e.g., digital signature) whose cryptographic binding to Composite Software identity is indistinguishable from the output obtained by another Composite Software on the same TOE, or (2) intercept and redirect data whose functionality (e.g., unwrapping) is intended to another Composite Software, and which is cryptographically bound to its identity.

The TOE ensures that by using this cryptographic capability, a Composite Software cannot impersonate another Composite Software, meaning that:

- The TOE must maintain the confidentiality of the provisioned private key and not reveal it to any Composite Software.
- The TOE must ensure that the TSF uniquely and specifically identifies every Composite Software.
- The cryptographic capability offered by the TOE must use the provisioned private key to ensure cryptographic binding between Composite Software identity and data it provides. It enables, depending on the cryptographic capability, (1) an entity outside the TOE to verify the cryptographic binding of the Composite Software identity on the cryptographic output or (2) the TOE to verify that input prepared and provided by an entity outside the TOE is intended for specified Composite Software identity, after the TOE successfully verified the cryptographic binding.

The provisioning of the key pair enables further use cases for the Composite Software developers: provisioning of devices (containing a SoC with 3S) during their manufacturing/packaging or over the air (OTA), in environments not covered by this evaluation after delivery of the TOE.

This optional package provides entities outside of the TOE, such as Composite Software developers, the necessary trust required to provision their Composite Software, after delivery of the TOE, in environments whose security is not covered by this evaluation or by evaluations applicable to use cases claimed by Composite Software developers.

7.7.1 Security Problem Definition

This package takes into account that the package assets are stored within the TOE and therefore the SFRs defined in the base PP are used for this purpose.

If the package assets are stored outside of the TOE, the Package for Passive External Memory or the Package for Secure External Memory shall be claimed in the security target.

This package re-uses the Package for Cryptographic Services. Therefore, this package must be included in the security target. This means that P.Crypto-Service and O.Crypto-Service must be instantiated with the algorithms using the Key pair.

7.7.1.1 Description of Assets

When this package is used, the additional assets of the TOE are:

- Provisioned asymmetric private key
- TSF data required for the cryptographic binding/verification between data and the identity of a Composite Software by the cryptographic capability

The additional assets covered by this package are considered part of the Root of Trust, and therefore all security requirements defined in the base profile extend to these assets.

7.7.1.2 Threats

The TOE shall avert the threat “Composite Software impersonation by cryptographic binding forgery (T.Impersonation-Forgery)” as specified below.

T.Impersonation-Forgery Composite Software impersonation by forgery

An attacker (A Composite Software, or a user outside the 3S) may attempt to impersonate another Composite Software, by using the cryptographic capability to forge output whose cryptographic binding to Composite Software identity is indistinguishable between two Composite Software instances.

Application Note 80. In the context of this threat, Impersonation-Forgery occurs when an entity outside of the 3S is unable to distinguish between the attacker’s cryptographic binding and the genuine cryptographic binding between Composite Software identity and data, which is cryptographically verified as originating from a genuine 3S instance.”

The TOE shall avert the threat “Composite Software Impersonation by interception of intended recipient (T.Impersonation-Interception)” as specified below.

T.Impersonation-Interception

Composite Software impersonation by interception of intended recipient

An attacker (A Composite Software, or an attacker outside its TOE utilizing a service it offers), may intercept data that was cryptographically bound by an entity outside the TOE to a 3S instance, and is intended to a Composite Software, and provide it to an another and unintended Composite Software on the same 3S instance.

7.7.1.3 Organisational Security Policy

Either the 3S Developer or the 3S Integrator shall apply the policy "Issuance of certificate (P.Iss-Cert)" as specified below.

P.Iss-Cert

Issuance of certificate

It shall be possible to process certificate signing request verifying the authenticity of the certificate signing request and issue the certificate by signing the certificate metadata and the public key associated to individual instance (3S key pair) or instances (3S group key pair) of the TOE.

Either the 3S Developer or the 3S Integrator shall apply the policy "Authorized subset of Composite Software (P.CompositeSW-Auth)" as specified below.

P.CompositeSW-Auth

Authorized subset of uniquely and specifically identified Composite Software

The 3S Developer or the 3S Integrator designate a subset of the Composite Software that is uniquely and specifically identified and is authorized to use TSF services involving usage of the provisioned key.

Application Note 81. In P.CompositeSW-Auth, specificity refers to the identities of authorized Composite Software, as ones that can be determined in advance, as opposed to a unique, but randomly generated identity.

Either the 3S Developer or the 3S Integrator shall apply the policy "Uniqueness of provisioned keys (P.KeyProv-Uniqueness)" as specified below.

P.KeyProv-Uniqueness

Uniqueness of the provisioned keys

Provisioned keys (generated in the TOE or injected) have a defined uniqueness. Provisioned keys are either be unique per 3S instance (3S key pair), or per group of 3S instances (3S group key pair).

The Organisational Security Policies included in the Package for Cryptographic Services shall be included in the Security Target.

7.7.1.4 Assumptions

The operational environment shall fulfil the assumption "Secure key provisioning during 3S in SoC manufacturing/packaging (A.KeyProvisioning)"as specified below.

A.KeyProvisioning

Secure key provisioning during 3S in SoC manufacturing/packaging

The provisioning of the cryptographic private material to the TOE is carried out in a secure environment during 3S in SoC manufacturing/packaging, before delivery of the TOE to OEM.

The operational environment shall fulfill the assumption "Secure lifecycle of keys used for provisioning (A.KeySecureLifecycle)" as specified below.

A.KeySecureLifecycle

Secure lifecycle of keys used for provisioning

When private keys used for provisioning 3S instances are generated outside the TOE and/or stored outside the TOE after provisioning, they are protected by security measures in the operational environment that ensure their confidentiality and integrity.

The operational environment shall fulfil the assumption "Delivery of the certificate signing request containing the public key to the certificate issuer (A.Delivery-CSR)"as specified below.

A.Delivery-CSR

Delivery of the certificate signing request containing the public key to the certificate issuer

The public cryptographic material associated to the provisioned private key shall be embedded within a certificate signing request and delivered to the certificate issuer. The embedding and delivery are carried out in a secure environment during 3S in SoC manufacturing/packaging.

7.7.2 Security Objectives

7.7.2.1 Security Objectives for the TOE

The TOE shall provide "Identification of a Composite Software (O.CompSW-Identification)" as specified below.

O.CompSW-Identification

Identification of a Composite Software

The TOE shall verify the specific identity of the Composite Software before performing operations that require the use of the provisioned private key on behalf of the Composite Software.

The TOE shall provide "Cryptographic binding of authorized Composite Software input to its identity (O.CompSW-Bind)" as specified below.

O.CompSW-Bind

Cryptographic binding of authorized Composite Software input to its identity

When producing output that requires cryptographic verification by an entity outside the TOE, the TOE cryptographic capability shall cryptographically bind, using the provisioned private key, the identity of authorized Composite Software to the input it provided.

The TOE shall provide "Verification of cryptographic binding (O. CompSW-VerifyBinding)" as specified below.

O.CompSW-VerifyBinding

Verification of cryptographic binding of authorized Composite Software input to its identity

When verifying data that was cryptographically bound by an entity outside the TOE, the TOE shall cryptographically verify the binding between input provided by an authorized Composite Software and its identity.

The Security Objectives for the TOE of this package, shall also include the Security Objectives for the TOE of the Package for Cryptographic Services. The term "purpose" shall be replaced by the cryptographic algorithm that are implemented by the TOE to be used with the provisioned key. The objective O.Crypto-Service shall be iterated for each cryptographic algorithm to support the mapping to the associated Security Functional Requirements.

7.7.2.2 Security Objectives for the TOE Environment

The operational environment of the TOE shall provide "Issuance of certificate (OE.Iss-Cert)" as specified below.

OE.Iss-Cert

Issuance of certificate

The 3S in SoC manufacturing/packaging environment shall provide the capability to process certificate signing request verifying the authenticity of the certificate signing request and issue the certificate by signing the certificate metadata and the public key associated to individual instance (3S key pair) or instances (3S group key pair) of the TOE.

The 3S integrator or 3S in SoC manufacturer shall provide "Authorized subset of Composite Software (OE.CompositeSW-Auth)" as specified below.

OE.CompositeSW-Auth

Authorized subset of Composite Software

The 3S Developer or the 3S Integrator shall designate a subset of uniquely and specifically identified Composite Software that is authorized to use TSF services involving usage of the provisioned key.

The operational environment of the TOE shall provide "Uniqueness of provisioned keys (OE.KeyProv-Uniqueness)" as specified below.

OE.KeyProv-Uniqueness

Uniqueness of provisioned keys

Provisioned keys (generated in the TOE or injected) shall have a defined uniqueness. Provisioned keys shall either be unique per 3S instance (3S key pair), or per group of 3S instances (3S group key pair)

The operational environment of the TOE shall provide "Secure key provisioning during 3S in SoC manufacturing/packaging (OE.KeyProvisioning)" as specified below.

OE.KeyProvisioning

Secure key provisioning during 3S in SoC manufacturing/packaging

The provisioning of the cryptographic private material to the TOE shall be carried out in a secure environment during 3S in SoC manufacturing/packaging, before delivery of the TOE to OEM.

Application Note 82. This package does not mandate specific ways to provision the private key material, and it is open depending on different TOE implementations. Keys may be generated outside the TOE by the 3S Developer or the 3S Integrator and then injected in the TOE, or the TOE may have a built-in key generation function that may be called during 3S in SoC manufacturing/packaging in order to create the key. The TOE may create a key pair; it may be injected a key pair, or it may be injected only the provisioned private key.

Application Note 83. The security of the environment and procedures associated to the provisioning of the private key material shall be assessed during ALC evaluation activities.

The operational environment of the TOE shall provide “Secure lifecycle of keys used for provisioning (OE.KeySecureLifecycle)” as specified below.

OE.KeySecureLifecycle

Secure lifecycle of keys used for provisioning

When private keys used for provisioning 3S instances are generated outside the TOE and/or stored outside the TOE after provisioning, they shall be protected by security measures in the operational environment that ensure their confidentiality and integrity.

The operational environment shall provide “Delivery of the certificate signing request containing the public key to the certificate issuer (OE.Delivery-CSR)” as specified below.

OE.Delivery-CSR

Delivery of the certificate signing request containing the public key to the certificate issuer

The public cryptographic material associated to the provisioned private key is embedded within a certificate signing request and delivered to the certificate issuer. The embedding and delivery are carried out in a secure environment during 3S in SoC manufacturing/packaging.

7.7.2.3 Security Objectives Rationale

	O.CompSW-Identification	O.CompSW-Bind	O.CompSW-VerifyBinding	OE.Iss-Cert	OE.CompositeSW-Auth	OE.KeyProv-Uniqueness	OE.KeyProvisioning	OE.KeySecureLifecycle	OE.Delivery-CSR
T.Impersonation-Forgery	X	X							

T.Impersonation-Interception			X						
P.Iss-Cert				X					
P.CompositeSW-Auth					X				
P. KeyProv-Uniqueness						X			
A.KeyProvisioning							X		
A.KeySecureLifecycle								X	
A.Delivery-CSR									X

Table 23: Mapping between security objectives and threats/policies

The justification for the threat "Composite Software impersonation by cryptographic binding forgery (T.Impersonation-Forgery)" is as follows: this threat is countered by the combination of two security objectives for the TOE:

- The objective "Identification of a Composite Software (O.CompSW-Identification)" ensures that Composite Software is identified before performing operations that require the use of the provisioned private key, preventing impersonation of another Composite Software by using the provisioned key on behalf of the latter.
- The objective "Cryptographic binding of authorized Composite Software input to its identity (O.CompSW-Bind)" ensures that the TOE cryptographic capability binds specific Composite Software identity to the input provided to an entity outside the TOE for its verification, hence it is possible to distinguish between the output generated by each Composite Software.

The justification for the threat "Composite Software impersonation by interception of intended recipient (T.Impersonation-Interception)" is as follows: this threat is countered by the security objective "Verification of cryptographic binding of authorized Composite Software input to its identity (O.CompSW-VerifyBinding)", that ensures that, for data generated by out-of-TOE entities, the TOE cryptographically verifies the binding between input provided and the identity of the Composite Software whose identity is bound to the input.

The justification for the policy "Issuance of certificate (P.Iss-Cert)" is as follows: this policy is directly implemented by the objective for the operational environment "Issuance of certificate (OE.Iss-Cert)", that ensures export of the public cryptographic material associated to the private provisioned key, in a secure environment and during 3S in SoC manufacturing/packaging.

The justification for the policy "Authorized subset of uniquely and specifically identified Composite Software (P.CompositeSW-Auth)" is as follows: this policy is directly implemented by the objective for the operational environment "Authorized subset of Composite Software (OE.CompositeSW-Auth)", that ensures that 3S Developer or the 3S Integrator designates a subset of the Composite Software as being authorized to use TSF services involving usage of the provisioned private key.

The justification for the policy "Uniqueness of provisioned keys (P.KeyProv-Uniqueness)" is as follows: this policy is directly implemented by the objective for the operational environment "Uniqueness of provisioned keys (OE.KeyProv-Uniqueness)", that ensures that uniqueness of provisioned keys per 3S instance or 3S instance group.

The justification for the assumption "Secure key provisioning during 3S in SoC manufacturing/packaging (A.KeyProvisioning)" is as follows: this assumption is directly upheld by the objective for the operational environment "Secure key provisioning during 3S in SoC manufacturing/packaging (OE.KeyProvisioning)”, that requires provisioning of the cryptographic key material to be done in a secure environment during 3S in SoC manufacturing/packaging.

The justification for the assumption “Secure lifecycle of keys used for provisioning (A.KeySecureLifecycle)” is as follows: this assumption is directly upheld by the objective for the operational environment “Secure lifecycle of keys used for provisioning (OE.KeySecureLifecycle)” which requires that keys used for provisioning which are generated and/or maintained outside the TOE after provisioning are protected with security measures of the environment that ensure their confidentiality and integrity.

The justification for the assumption “Delivery of the certificate signing request containing the public key to the certificate issuer (A.Delivery-CSR)” is as follows: this assumption is directly upheld by the objective for the operational environment “Delivery of the certificate signing request containing the public key to the certificate issuer (OE.Delivery-CSR)”, which requires that the public cryptographic material associated to the provisioned private key is embedded within a certificate signing request and delivered to the certificate issuer, all done in a secure environment during in SoC manufacturing/packaging.

7.7.3 Extended Component Definition

In order to address the security aspects related to cryptographically verify the binding of data generated by out-of-TOE entities with the public certificate associated to the provisioned private key, the extended family FDP_CDP: Data protection by cryptographic binding verification has been added.

7.7.3.1 Definition of the family FDP_CDP

Family behaviour

This family defines functions provided by the TOE to perform cryptographic verification of the binding between data that has cryptographically bound by an out-of-TOE entity, and the identity of a specific authorized Composite Software. This binding is performed with the public certificate associated to the provisioned private key described in this package.

Component levelling

FDP_CDP: Data protection by cryptographic binding verification
1

FDP_CDP.1 Data protection by cryptographic binding verification, provides the capability to cryptographically verify the binding of data bound by out-of-TOE entities with the identity of specific authorized Composite Software.

Management **FDP_CDP.1**

There are no management activities foreseen.

Audit **FDP_CDP.1**

There are no actions defined to be auditable.

FDP_CDP.1 **Data protection by cryptographic binding verification**

Hierarchical to: No other components.

Dependencies: FCS_COP.1 Cryptographic operation

FDP_CDP.1.1 The TSF shall perform cryptographic verification of the binding between input data generated by an entity outside the TOE and the identity of specific authorized composite user software.

7.7.4 IT Security Requirements

7.7.4.1 SFRs for the TOE

The TOE shall meet the requirement "User identification before any action (FIA_UID.2) Refined", as follows:

FIA_UID.2 **User identification before any action (Refined)**

Hierarchical to: No other components.

Dependencies: None

FIA_UID.2.1 The TSF shall require ~~each user~~ **the Composite Software** to be successfully identified before allowing ~~any other TSF-mediated actions~~ **usage of the provisioned cryptographic asymmetric material** on behalf of ~~that user~~ **the Composite Software**.

Application Note 84. The intent of this SFR is to verify the identity of the Composite Software before performing, on its behalf, an operation that involves usage of the provisioned private key. If the TOE is provisioned with more than one key, the TSF shall verify the identity of the Composite Software upon request of usage of each of the keys on its behalf. Moreover, the Security Target must indicate (e.g., in the TOE Summary Specification), whether the TOE is provisioned with a single key or with a group of keys.

The TOE shall meet the requirement "Data Authentication with Identity of Guarantor (FDP_DAU.2/CS) Refined", as follows:

FDP_DAU.2/CS **Data Authentication with Identity of Guarantor (Refined)**

Hierarchical to: FDP_DAU.1 Basic Data Authentication

Dependencies: FIA_UID.1 Timing of identification

FDP_DAU.2.1/CS The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of an authorized subset of the Composite Software[56].

FDP_DAU.2.2/CS The TSF shall provide out-of-TOE entities[57] with the ability to verify evidence of the validity of the indicated information and the identity of the ~~user~~ **Composite Software** that generated the evidence.

The TOE shall meet the requirement "Data protection by cryptographic binding verification (FDP_CDP.1)", as follows:

FDP_CDP.1 **Data protection by cryptographic binding verification**

[56] [assignment: *list of objects or information types*]

[57] [assignment: *list of subjects*]

Hierarchical to:	No other components.
Dependencies:	FCS_COP.1 Cryptographic operation
FDP_CDP.1.1	The TSF shall perform cryptographic verification of the binding between input data generated by an entity outside the TOE and the identity of specific authorized composite user software.

Application Note 85. The Security Target author shall include the relevant iterations of the FCS_COP.1/Iteration requirement from the "Package for Cryptographic Services", in order to ensure that the necessary cryptographic algorithms are included in the TSF for cryptographic verification of binding generated by out-of-TOE entities, using the corresponding public cryptographic material. By doing this, it will be ensured that the TSF has the capability to use the provisioned private key to verify the input data, and also the dependency of this SFR on FCS_COP.1 can be satisfied.

7.7.4.2 Rationale for the SFRs

Table **24** 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.CompSW-Identification	FIA_UID.2
O.CompSW-Bind	FDP_DAU.2/CS
O.CompSW-VerifyBinding	FDP_CDP.1, FCS_COP.1/Iteration

Table 24: Mapping between Objectives and SFRs for the Asymmetric Provisioning Package with Key Generation

O.Crypto-Service and O.RND are not included in this mapping because they are part of the Package for Cryptographic Services or the base PP respectively.

The justification related to the security objective “Identification of a Composite Coftware (O.CompSW-Identification)” is as follows: O.CompSW-Identification is met by FIA_UID.2, which requires identification of the Composite Software before allowing to perform any operations requiring usage of the provisioned key on behalf of the Composite Software.

The justification related to the security objective “Cryptographically binding of authorized Composite Software input to its identity (O.CompSW-Bind)” is as follows: O.CompSW-Bind is met by FDP_DAU.2/CS which requires the TSF to generate an evidence guaranteeing the validity of the authorized Composite Software, in a way that out-of-TOE entities can verify the identity of the Composite software based on the generated evidence, which is cryptographically bound to the identity of the Composite Software.

The justification related to the security objective “Verification of cryptographic binding of authorized Composite Software input to its identity (O.CompSW-VerifyBinding)” is as follows O.CompSW-VerifyBinding is met by FDP_CDP.1, which requires the TOE to be able to verify the cryptographic binding between the input data provided by out-of-TOE entities and the identity of a specific authorized Composite Software. Moreover, FCS_COP.1/Iteration contributes to meet this objective, as it provides the cryptographic operations needed for the verification of cryptographic binding.

Application Note 86. The ST author shall modify the above table and rationale to reference the instances of FCS_COP.1/Iteration of the Package for Cryptographic Services that are included in the ST in relation with the cryptographic operations that need to be done to verify or unwrap externally-generated data with the provisioned private key.

7.7.4.3 Dependencies of the SFRs

Requirement	Dependencies	Satisfied Dependencies
FIA_UID.2	No dependency	-
FDP_DAU.2/CS	FIA_UID.1	FIA_UID.2
FDP_CDP.1	FCS_COP.1	FCS_COP.1/Iteration

Table 25: Overview of Dependencies of the SFRs for the Asymmetric Provisioning Package with Key Generation

Application Note 87. The Security Target author shall indicate in the ST the relevant iterations of FCS_COP.1/Iteration requirement from the "Package for Cryptographic Services” that serve to meet the dependency of FDP_CDP.1 on FCS_COP.1.

8 References and Acronyms

8.1 References

[3] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, CC:2022, Revision 1, November 2022, CCMB-2022-11-001

[4] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, CC:2022, Revision 1, November 2022, CCMB-2022-11-002

[5] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, CC:2022, Revision 1, November 2022, CCMB-2022-11-003

[6] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 4: Framework for the specification of evaluation methods and activities, CC:2022, Revision 1, November 2022, CCMB-2022-11-004

[7] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 5: Pre-defined packages of security requirements, CC:2022, Revision 1, November 2022, CCMB-2022-11-005

[8] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, CC:2022, Revision 1, November 2022, CCMB-2022-11-004

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

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

[11] SOG-IS Crypto Evaluation Scheme Agreed Cryptographic Mechanisms latest version

[12] 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 referenceBSI-CC-PP-0084-2014.

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

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

[15] Supporting Document: The Application of CC to Integrated Circuits, March 2009, Version 3.0, Revision 1

[16] Supporting Document Guidance: Smartcard Evaluation, February 2010, Version 2.0

[17] Supporting Document: Security Architecture requirements (ADV_ARC) for smart cards and similar devices, Jan 2012, Version 2.0

[18] Supporting Document: Application of Attack Potential to Smartcards June 2020, Version 3.1

[19] Supporting Document: Composite product evaluation for Smart Cards and similar devices, May 2018, Version 1.5.1

[20] Supporting Document: Minimum site security requirements, Feb. 2020, Version 3.0

8.2 Acronyms

Acronym	Definition
CC	Common Criteria
3S	Security Sub-System
EAL	Evaluation Assurance Level
FPGA	Field Programmable Gate Array
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 [12], 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 26: Comparison between threats in [12] and this PP

The OSP in this PP is taken over and renamed compared to the OSPs in BSI-CC-PP-0084-2014 [12], 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 same scope. The 3S may be integrated in SoCs of different vendors. Therefore, the requirement is extended to ensure a unique identification across all vendors; for details, see OE.Secure-Initialisation.

Table 27: Comparison between OSPs in [12] and this PP

The assumptions in this 3S PP are slightly adapted compared to the assumptions in BSI-CC-PP-0084-2014 [12], 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 3S.

Table 28: Comparison between assumptions in [12] and this PP

The Security Objectives in this 3S PP are extended compared to the Security Objectives in BSI-CC-PP-0084-2014 [12], 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.

Security Objectives in PP0084	Security Objectives in this PP	Rationale
	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.
	OE.Secure-Initialisation	Additional objective for the environment, because the unique identification must be ensures throughout all 3S, even if the initialisation may be performed by different vendors.
	OE.Packaging-Requirement	Additional objective for the environment, because the packaging is not defined in **BSI-CC-PP-0084-2014**.

Table 29: Comparison between Security Objectives in [12] and this PP

The security requirements in this PP are a superset of the security requirements in the PP BSI-CC-PP-0084-2014 [12], to which conformance is claimed. The security objectives of the TOE are mapped to the same SFRs in both Protection Profiles with the following extensions:

Security Functional Requirements in PP0084	Security Functional Requirements 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”	FDP_ITT.1/3S “Basic internal transfer protection” FPT_ITT.1/3S “Basic internal TSF data transfer protection” FDP_IFC.1/3S “Subset information flow control”	For “O.Leak-Inherent” as defined in the base Protection Profile the three SFRs FDP_ITT.1, FPT_ITT.1 and FDP_IFC.1 with the unique identifier /3S. For configurations with secure external memory a dedicated objective regarding the protection of the external memory against leakage is included in this functional package. Since the data exchange between the 3S and the secure external memory is under control of the TOE with firmware and software, the SFP is adapted.
	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”.

Security Functional Requirements in PP0084	Security Functional Requirements in this PP	Rationale
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 30: Comparison between the SFRs in [12] 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 [9]. These documents describe predefined classes PTG.2, PTG.3 and DRG.4 of random number generators (cf. [10]) 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[58] 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

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

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[59].

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[60].

Application Note 88. 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. [9]).

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 [13] and [14].

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 [13] and NIST Special Publication 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation, January 2018 [14]. The draft recommendation for entropy sources [14] 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. [13] defines hybrid deterministic RNG designs. Note [13] 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 [14] 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.

[59] [assignment: *list of security capabilities*]

[60] [assignment: *a defined quality metric*]

FCS_RNG.1.1/ES The TSF shall provide a *physical*[61] 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*][62].

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]]*[63].

The clause (ES.3) may describe conditioning components implementing NIST approved or non-approved cryptographic functions, which are optional in [14]. 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 $0 \le \varepsilon \le 2^{-64}$.

If the TOE shall implement hybrid random number generator of the TOE complying to [13] 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.

FCS_RNG.1.1/HD The TSF shall provide a *hybrid deterministic*[64] random number generator that implements: [selection: *CTR_DRBG, Hash_DRBG, HMAC_DRBG*] as defined in NIST Special Publication 800-90A [13][65].

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*][66].

For details of the security capabilities and the security bits as quality metric of the random number output, see NIST Special Publication 800-90A [13].

9.3 SFR changes according to CC:2022

This section provides a summary of the changes applied to the SFRs according to CC:2022, Revision 1. The SFRs included in Table 31 were instantiated in this PP according to BSI-CC-PP-0084-2014 as strict conformance is claimed to it. However, some of the SFRs taken from [10] have been adapted to CC:2022 Part 2 [2], having into account the differences between both instantiations as can be seen in

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

[62] [assignment: *list of security capabilities*]

[63] [assignment: *a defined quality metric*]

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

[65] [assignment: *list of security capabilities*]

[66] [assignment: *a defined quality metric*]

Table 31. Some of them were defined in [10] as extended components, but CC:2022 has included them in [2].

SFR	Changes
FCS_RNG.1	There are no differences between FCS_RNG.1 instantiation in BSI-CC-PP-0084-2014 and the instantiation in CC:2022 Revision 1. The requirement is equally defined. Therefore, it has been removed from Extended Components section and only has been included in the Security Functionals Requirements for the TOE.
FMT_LIM.1	The differences between FMT_LIM.1 instantiation in BSI-CC-PP-0084-2014 and the instantiation in CC:2022 Revision 1 are limited to the terminology and they have no impact. Therefore, it has been removed from Extended Components section and only has been included in the Security Functionals Requirements for the TOE.
FMT_LIM.2	The differences between FMT_LIM.2 instantiation in BSI-CC-PP-0084-2014 and the instantiation in CC:2022 Revision 1 are limited to the terminology and they have no impact. Therefore, it has been removed from Extended Components section and only has been included in the Security Functionals Requirements for the TOE.
FDP_SDC.1	The FDP_SDC.1 instantiation in BSI-CC-PP-0084-2014 and the instantiation in CC:2022 Revision 1 are equivalent. Therefore, it has been removed from Extended Components section and only has been included in the Security Functionals Requirements for the TOE.
FPT_INI.1	The FPT_INI.1 instantiation in BSI-CC-PP-0084-2014 and the instantiation in CC:2022 Revision 1 has considerably changed. On the one hand, the CC:2022 instantiation adds the implementation of an initialization function which is self-protected for integrity and authenticity. On the other hand, the TOE initialization function in CC:2022 add the possibility of initialize the TOE with either reduced functionality, signalling error state or a list of actions provided by the ST author. Lastly, the CC:2022 instantiation allows to define the available methods to interact with the TSF during the initialization process. Therefore, it has been removed from Extended Components section and only has been included in the Security Functionals Requirements for the TOE.
FCS_CKM.6	FCS_CKM.4 is obsolete in CC:2022. It is replaced by FCS_CKM.6. FCS_CKM.6 in CC:2022 allows to indicate the cryptographic keys which shall be destroyed and the reason to destroy them. This SFR is only included in Package for Cryptographic Services described in section 7.4.
FCS_COP.1	FCS_COP.1 instantiation in CC:2022 removes the dependency with FCS_CKM.4 (and FCS_CKM.6), adds an optional dependency with FCS_CKM.5 and adds a mandatory dependency with FCS_CKM.3. The dependencies are not satisfied in this PP, but it is indicated that the ST author is responsibly of determine if the dependencies are satisfied or not, and if they are satisfied indicate how the SFRs are implemented. This SFR is only included in Package for Cryptographic Services described in section 7.4.

Table 31: SFR changes in CC:2022.

EUROSMART

The Voice of the Digital Security Industry

www.eurosmart.com

@Eurosmart_EU

@Eurosmart