EUROSMART

The Voice of the Digital Security Industry

Security IC Platform Protection Profile including Functional Packages

Version 1.0 (CC:2022 Update v0.3)

Date: 13/08/2025

developed by

Infineon Technologies AG

NXP Semiconductors Germany GmbH

STMicroelectronics

Thales

Registered and Certified by
Bundesamt für Sicherheit in der Informationstechnik (BSI)
under the reference BSI-CC-PP-0084-2014.

This page is intentionally left blank.

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 PP Identification

1 The identification of this Protection Profile (PP) consists of the title, version, date and certificate number presented hereafter.

Title:	Security IC Platform Protection Profile and optional functional packages
Version:	Version 1.0 (CC:2022 Update v0.3)
Date:	13/08/2025
Sponsored by:	Infineon Technologies AG, NXP Semiconductors, STMicroelectronics, and Thales
Technical editors:	Internet of Trust, 77 avenue Niel, 75017 Paris, France
Certified by:	Bundesamt für Sicherheit in der Informationstechnik (BSI) under certificate number BSI-CC-PP-0084-2014

1.2 PP Overview

2 This document has been developed based on:

[14] Eurosmart, Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, January 2014, BSI-CC-PP-0084-2014

[15] Eurosmart, Smartcard Integrated Circuit Platform Augmentations, Version 1.00, March 2002.

It is an update of [14] which incorporates an evolution of the functional package “Area-based Memory Access Control” defined in [15] and is conformant to CC:2022.

3 This document consists of

- the core PP for security integrated circuits (Security IC), consisting of the security problem definition (SPD), security objectives and security requirements that apply to all targets of evaluation (TOE) of this type, and
- five optional functional packages for Security ICs with extended security functionality:
  - Package “Authentication of the Security IC”,
  - Package “Loader dedicated for usage in secured environment only”,
  - Package “Loader dedicated for usage by authorized users only”,
  - Package “Cryptographic services”,
  - Package “Address-based access control”.

4 This document contains numbered “Application Notes” which provide additional information about specific topics; some of them include requirements, i.e. “shall”

statements.

1.3 TOE Overview

1.3.1 TOE Type

5 The TOE type is a Security IC which is composed of a processing unit, security components, I/O ports (contact, contactless, or similar interfaces like USB) and volatile and non-volatile memories.

6 The Security IC may include IC Dedicated Software if it is delivered by the IC Manufacturer, which is often used for testing purposes during production only but may also provide additional services, e.g. to facilitate usage of the hardware and/or to provide additional services (for instance in the form of a library). In addition, the Security IC may also comprise hardware to perform testing.

7 The TOE comprises

- the circuitry of the IC (hardware including the physical memories),
- if applicable, the IC Dedicated Software,
- the Configuration Data and the Initialisation Data related to the behaviour of the security functionality[1],
- the associated guidance documentation.

1.3.2 TOE Description

8 Figure 1 depicts a typical Security IC, which is composed of a processing unit, security components, I/O ports, volatile and non-volatile memories and optional cryptographic processors. The countermeasures against physical tampering (e.g. shields), environmental stress (e.g. sensors) and other attacks (cf. 3.2) provided by the Security IC but not directly related to other blocks are shown in a security circuitry block.

[1] The data may also be coded in specific circuitry of the IC.

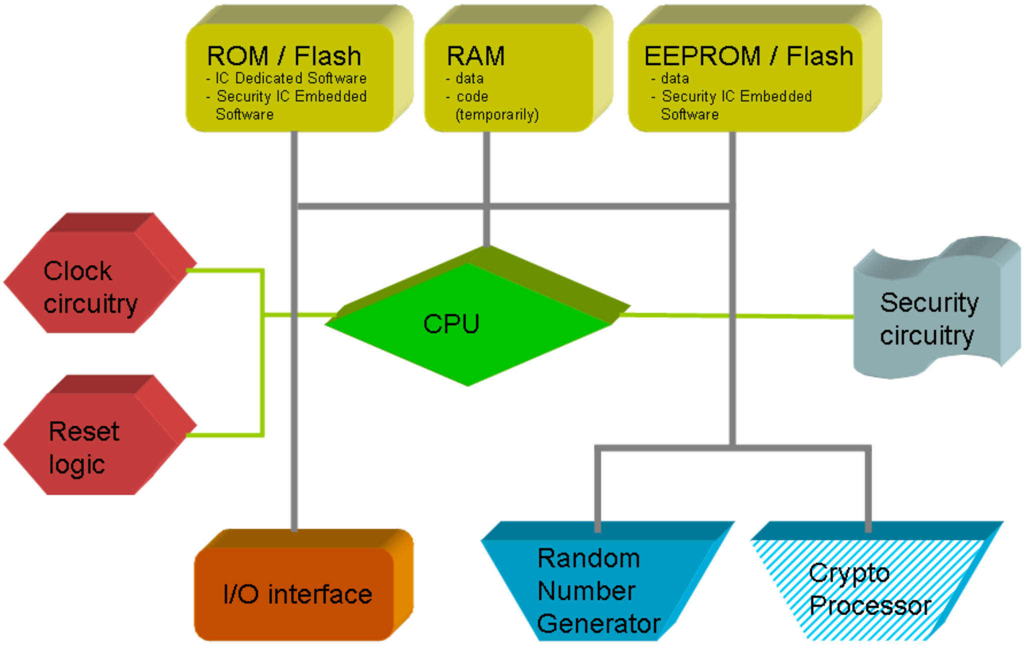


Figure 1: Typical Security IC

9 The TOE is designed, produced and/or generated by the TOE Manufacturer.

10 The Configuration Data and Initialisation Data related to the IC Dedicated Software and the behaviour of the security functionality are coded in non-volatile non-programmable memories (ROM), in non-volatile programmable memories (NVM), in specific circuitry or a combination thereof.

11 The IC Dedicated Test Software is only used to support testing of the TOE during production and does not provide security functionality to be used after TOE Delivery. Therefore, this software (or parts of it) is seen only as a test tool though being delivered as part of the TOE. However, it shall be verified that it cannot be abused after TOE Delivery.

12 In contrast, the IC Dedicated Support Software does provide functions after TOE Delivery. Therefore, during the evaluation it is treated as any other part of the TOE. The IC Dedicated Support Software may be stored in ROM or in NVM. It may be delivered as source code or libraries in addition to the hardware.

13 The TOE is intended to be used for a Security IC-based product, independent of the physical interface and the way it is packaged. Note that the Security IC is usually packaged. However, this PP does not specify the way it is packaged.

14 A Security IC Product may include other optional elements such as specific hardware components, batteries, capacitors or antennae, which are not in the scope of this PP and can be defined in the Security Target (ST).

15 The Composite Product comprises

- the Security IC and, if delivered and available to the user of the Composite Product, the IC Dedicated Support Software,
- the Security IC Embedded Software, more specifically,

  - the hard-coded Security IC Embedded Software (normally stored in ROM),
  - the soft-coded Security IC Embedded Software (normally stored in NVM), and
- the user data of the Composite TOE (especially the personalisation data and other data generated and used by the Security IC Embedded Software).

16 Typically, the TOE Manufacturer neither designs the Security IC Embedded Software nor generates the user data of the Composite TOE. This is user data from the point of view of the TOE.

17 The Security IC Embedded Software can be stored in ROM and/or in NVM, refer to 11.1. Typically,

- the IC Manufacturer installs all or the main part of the Security IC Embedded Software in ROM or NVM during the manufacturing of the TOE, and
- the Composite Product Integrator only installs supplements for the Security IC Embedded Software by means of the Security IC Embedded Software itself in NVM.

18 The installation of the Security IC Embedded Software and the user data of the Composite Product in NVM by means of IC Dedicated Support Software is addressed through the following functional packages:

- Loader Package 1 defined in 7.2.1, if the installation occurs up to Personalisation Phase (Phase 6) included, and
- Loader Package 2 defined in 7.2.2, if the installation occurs in the TOE Operational Usage Phase (Phase 7).

19 All data managed by the Security IC Embedded Software is called user data of the Composite TOE. In addition, Pre-personalisation Data (refer to 11.1) may contain TSF data and user data of the TOE.

20 Further terms are explained in the Glossary (refer to 8).

1.3.3 TOE Major Security Features

21 The security characteristics of a Security IC can be summarised as the ability to defend against attempts to commit fraud, gain unauthorised access to data or take control of a system through the Security IC by attackers with high attack potential as defined in [10]. Therefore, the major security features of a Security IC are to

- maintain the integrity of the content of the Security IC memories and the confidentiality of the content of protected memory areas as required by the application(s) the Security IC is built for, and
- maintain the correct execution of the software residing on the Security IC.

22 A Security IC may also protect data in other memory areas even if not required by the security functional requirements (SFRs), e.g. in ROM. Effective user data protection requires that the Security IC especially maintains the integrity of its TSF and TSF data and their confidentiality if necessary.

1.3.4 TOE Usage

23 Security ICs are commonly integrated into products such as smart cards, USB tokens and other devices to protect their information and services.

24 Protected information generally includes secret or integrity-sensitive data, such as personal identification numbers, balance values and personal data files, as well as access rights and cryptographic keys needed for accessing to the protected information and using the services provided by the product.

25 Examples of Security IC-based products include electronic passports, payment cards and authentication tokens. These products operate in a wide range of environments, and once issued, they may be stored and used globally at any time. Typically, no control can be exercised over the Security IC or its operational environment after deployment of the embedding product.

1.3.5 TOE Life Cycle

1.3.5.1 Phases

26 The complex development and manufacturing processes of a Composite Product can be separated into seven distinct phases. The phases 2 and 3 of the Composite Product life cycle cover the IC development and manufacturing:

- IC Development (Phase 2):
  - IC design,
  - IC Dedicated Software development,
- IC Manufacturing (Phase 3):
  - Integration and photomask fabrication,
  - IC production,
  - IC testing,
  - Initialisation, and
  - Pre-personalisation if necessary.

27 The Composite Product life cycle phase 4 can be included in the evaluation of the IC as an option:

- IC Packaging (Phase 4):
  - Security IC packaging (and testing),
  - Pre-personalisation if necessary.

28 In addition, four important stages shall be considered in the Composite Product life cycle:

- Security IC Embedded Software Development (Phase 1),

- Composite Product finishing process, preparation and shipping to the personalisation line for the Composite Product (Composite Product Integration Phase 5),

- Composite Product personalisation and testing stage where the user data of the Composite TOE is loaded into the Security IC's memory (Personalisation Phase 6),

- Composite Product usage by its issuers and consumers (Operational Usage Phase 7) which may include loading and other management of applications in the field.

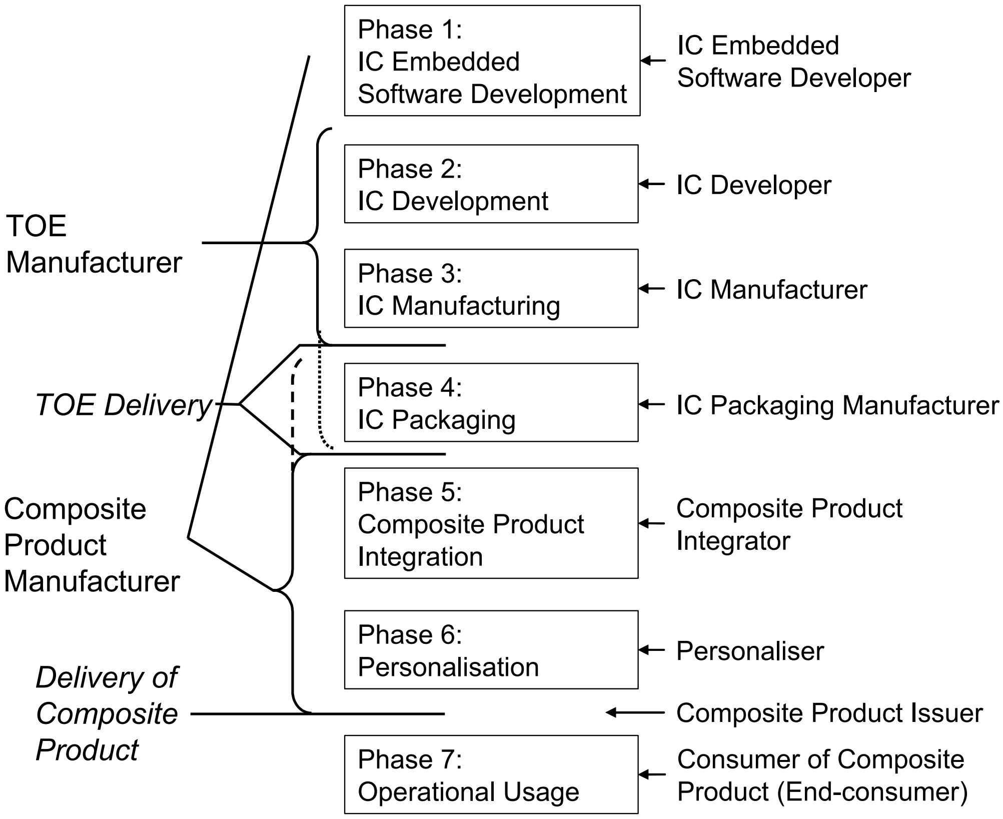


Figure 2: Definition of "TOE Delivery" and responsible Parties

29 The Security IC Embedded Software is developed outside the TOE development in Phase 1. The TOE is developed in Phase 2 and produced in Phase 3. Then the TOE can be delivered in form of wafers or sawn wafers (dice). The TOE can also be delivered in form of packaged products. In this case the corresponding assurance requirements of this PP for the development and production of the TOE not only pertain to Phase 2 and 3 but to Phase 4 in addition. Refer to the life cycle description in 11.1.2.

30 In the following, the term "TOE Delivery" (refer to Figure 2) is used exclusively to indicate

- after Phase 3 (or before Phase 4) if the TOE is delivered in form of wafers or sawn wafers (dice) or
- after Phase 4 (or before Phase 5) if the TOE is delivered in form of packaged products.

31 This PP uses the term “TOE Manufacturer” (refer to Figure 2) which includes the following roles:

- IC Developer (Phase 2) and
  IC Manufacturer (Phase 3)

  if the TOE is delivered after Phase 3 in form of wafers or sawn wafers (dice) or

- IC Developer (Phase 2),
  IC Manufacturer (Phase 3) and
  IC Packaging Manufacturer (Phase 4)

  if the TOE is delivered after Phase 4 in form of packaged products.

32 Hence the “TOE Manufacturer” comprises all roles from Phase 2 and before “TOE Delivery”. Upon “TOE Delivery” another party takes over the control of the TOE. This PP defines assurance requirements for the TOE’s development and production environment up to “TOE Delivery”. Refer to Figure 2.

33 This PP uses the term “Composite Product Manufacturer” to encompass all roles involved in the Composite Product life cycle, excluding TOE development and manufacturing, and the End-consumer as the final user of the Composite Product (refer to Figure 2). These roles are the following:

- Security IC Embedded Software development (Phase 1)
- IC Packaging Manufacturer (Phase 4)
  if the TOE is delivered after Phase 3 in form of wafers or sawn wafers (dice)
- Composite Product Manufacturer (Phase 5) and
  Personaliser (Phase 6).

Application Note 1: The ST shall explicitly state whether (i) TOE Delivery occurs after Phase 3 only or (ii) after Phase 4 as well. This can be done by using the relevant information from the paragraphs above. A detailed description of the life cycle is given in 11.1.

Application Note 2: If the TOE provides functionality to be used after TOE Delivery this is part of the IC Dedicated Support Software. Then such functions shall be specified in the ST of the actual TOE. Revise the above paragraphs in the ST to make clear if the TOE comprises IC Dedicated Support Software (e.g. a loader for the NVM).

1.3.5.2 Life Cycle versus Scope and Organisation of this PP

34 The whole life cycle of the Composite Product will be considered during evaluations using this PP as far as the TOE Manufacturer is directly involved. The details are given in terms of refinements of the Common Criteria (CC) assurance components which cover the development and production processes of the TOE.

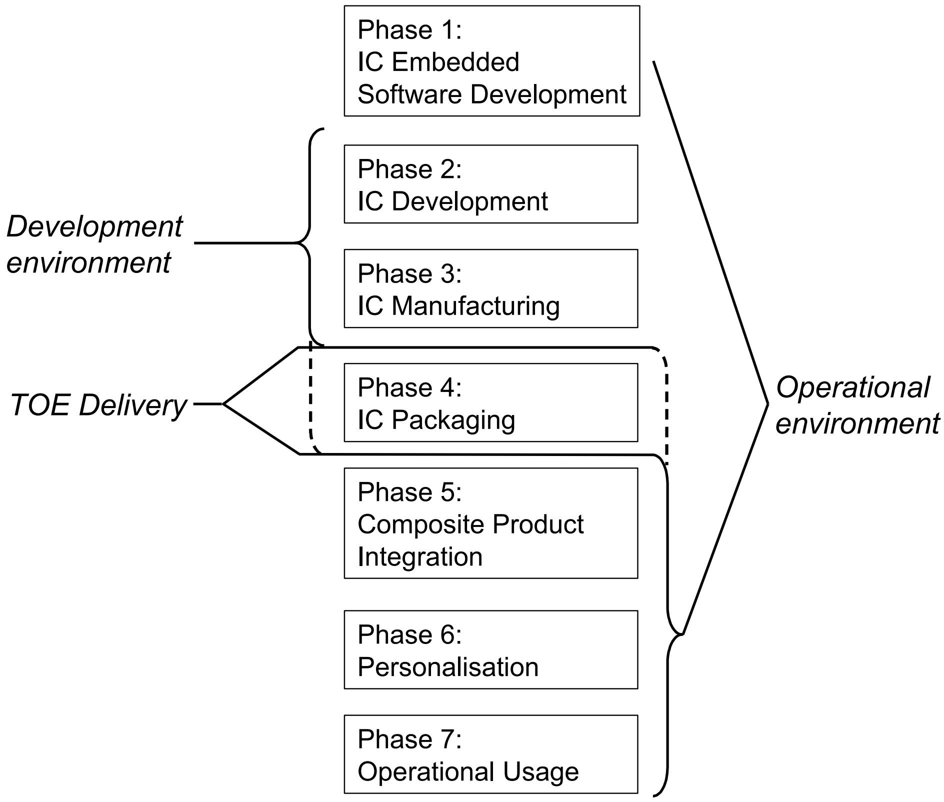


Figure 3: Development environment and Operational environment

35 The scope of the assurance components referring to the TOE's life cycle is limited to Phases 2 and 3, which are under the control of the TOE Manufacturer.

36 The IC Packaging and Testing in Phase 4 may be included in the scope if the TOE Manufacturer delivers packaged TOE, refer to the dashed line in Figure 3.

37 All procedures within these phases are addressed by the PP. This includes the interfaces to the other phases where information and material is being exchanged. The Composite Product Manufacturer and the TOE Manufacturer interact and may exchange critical information. Therefore, CC assurance requirements are refined in 6.2.2 to ensure that this PP exactly reflects the requirements for the exchange of information and material between the TOE Manufacturer and the Composite Product Manufacturer.

38 In particular, the CC assurance requirements ALC_DEL (delivery) and AGD_PRE are refined. So, the details regarding development of the Security IC Embedded Software, secure delivery and receipt of TOE are addressed.

39 It may be necessary to state security objectives for other parties in the ST if they use security critical information of the TOE Manufacturer. However, it cannot be assessed during an evaluation of the TOE whether these security objectives for the TOE environment are met. Consequently, these requirements shall be considered during the evaluation of the Security IC Embedded Software or Composite Product.

40 For assumptions regarding the usage of the TOE (its environment) made in this PP refer to 3.4.

Application Note 3: The TOE may provide functions supporting the Security IC's life cycle (for instance secure/authentic delivery). In this case the corresponding requirements shall be specified in the ST in terms of security objectives and functional requirements. This is visualised in Figure 3.

41 This approach of Security IC life cycle versus PP requirements is visualised in Figure 3. Additional requirements may be chosen to correctly interface to a PP including the Security IC Embedded Software.

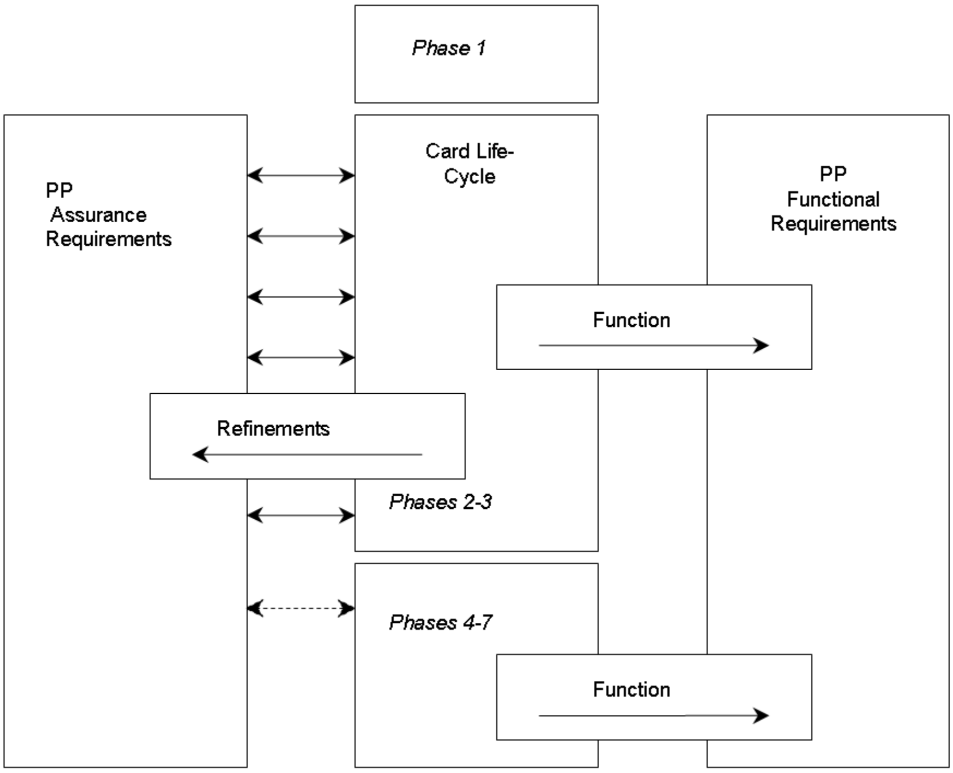


Figure 1: Security IC Life Cycle versus PP Requirements

2 Conformance Claims

2.1 CC Conformance Claim

42 This PP claims conformance to CC:2022 Revision 1, cf. [1], [2], [3], in the following way:

- CC Part 2 extended,
- CC Part 3 conformant.

43 The extended security functional requirements are defined in chapter 5.

44 The evaluation methodology [5] has been considered.

2.2 PP Conformance Claim

45 This PP does not claim conformance to any other PP.

2.3 Package Claim

2.3.1 Functional Package Claim

46 This PP claims conformance to the functional packages defined in this document:

- Package “Authentication of the Security IC”, cf. 7.1,
- Packages for Loader:
  - Package Loader 1 “Loader dedicated for usage in secured environment only”, cf. 7.2.1,
  - Package Loader 2 “Loader dedicated for usage by authorized users only”, cf. 7.2.2,
- Package “Cryptographic Services”, cf. 7.3,
- Package “Address-based Access Control”, cf. 7.4.

47 These packages are optional and independent from each other. This means that a PP-conformant ST can include any of them if the underlying functionality is supported by the TOE.

2.3.2 Assurance Package Claim

48 The minimum assurance level for this Protection Profile is EAL4 augmented with ALC_DVS.2, ALC_FLR.2 and AVA_VAN.5 (refer to 6.2 for details).

2.4 Conformance Statement

49 This PP requires strict conformance of an ST or PP claiming conformance to it.

2.5 Conformance Claim Rationale

2.5.1 General

50 The functional packages defined in this document are independent from each other. Each of them addresses a well-defined additional TOE functionality which does not affect the functionality defined in the core PP or in the other packages. Therefore, a PP-conformant ST may include none or any subset of functional packages. For instance,

- An ST that includes Package Loader 2 may also include the Package “Authentication of the Security IC” for authentication of the TOE as end point of the trusted channel.
- An ST may include both Packages Loader 1 and Loader 2, to cover all types of environments. This can be performed by different loaders, or by the same loader if limitation of capabilities and availability is provided by the TOE.

2.5.2 Package “Authentication of the Security IC"

51 This package extends the core PP. It addresses optional functionality that allows to authenticate the TOE to external users:

- The package extends the SPD in the core PP by defining a threat in which an attacker impersonates the legitimate TOE using a non-genuine device. This threat is specific to the additional functionality and does not undermine the SPD defined in the core PP.
- The package introduces the objective that the IC (the TOE) shall be able to authenticate itself to external entities. This objective is specific to the additional functionality and does not contradict the objectives defined in the core PP.
- The requirement FIA_API.1 fulfils that objective by defining an authentication mechanism to prove the TOE's identity, and it does not contradict with any SFRs in the core PP.

2.5.3 Package Loader 1 “Loader dedicated for usage in secured environment only"

52 This package extends the core PP. It addresses optional functionality allowing the secure loading of software or data into the TOE only during controlled and trusted phases of the life cycle (e.g., before TOE Delivery to End-customer):

- The package extends the SPD in the core PP by specifying a scenario where loading of the Security IC Embedded Software, user data of the Composite Product or IC Dedicated Support Software can only occur in a controlled and secured environment. The SPD extension is specific to the additional functionality and does not undermine the SPD defined in the core PP.
- The package introduces objectives to ensure that any loading process is performed under restricted conditions to prevent unauthorized manipulation or disclosure of software and/or data. These objectives are specific to the additional functionality and do not contradict the objectives defined in the core PP.
- The requirements FMT_LIM.1/Loader and FMT_LIM.2/Loader contribute to the package’s objectives by enforcing strict controls on the loader’s operation and deployment. These requirements align with and do not contradict the SFRs in the core PP.

2.5.4 Package Loader 2 “Loader dedicated for usage by authorized users only "

53 This package extends the core PP. It addresses optional functionality enabling the secure loading of software or data into the TOE after delivery, restricted to operations performed by authorized users under authentication conditions:

- The package extends the SPD in the core PP by specifying a scenario where the loader functionality is restricted to authorized users. The SPD extension is specific to the additional functionality and does not undermine the SPD defined in the core PP.

- The package introduces objectives to ensure that the loader is accessible only to authenticated users to prevent unauthorized manipulation or disclosure. These objectives are specific to the additional functionality and do not contradict the objectives defined in the core PP.

- The SFRs for user authorization contribute to the package's objectives by enforcing strict identity checks prior to any loading process. These requirements align with and do not contradict the SFRs in the core PP.

2.5.5 Package “Cryptographic Services"

54 This package extends the core PP. It addresses optional functionalities that provide standard cryptographic operations for use by the Security IC Embedded Software:

- The package extends the SPD in the core PP by defining an OSP for cryptographic services to ensuring data protection within the IC. The SPD extension is specific to the additional functionality and does not undermine the SPD defined in the core PP.

- The packages introduce an objective to guaranteeing that cryptographic mechanisms provided to the Security IC Embedded Software are hardware-based. This objective is specific to the additional functionality and do not contradict the objectives defined in the core PP.

- The SFRs fulfil the objective by enforcing standardized cryptographic functionality. These SFRs align with and do not contradicting the SFRs of the core PP.

2.5.6 Package “Address-based Access Control"

55 This package extends the core PP. It addresses optional functionality that enables the TOE to enforce address-based access restrictions. The functionality supports memory partitioning and prevents unauthorized memory access at least:

- The package extends the SPD in the core PP by defining a new threat which covers accidental or deliberate unauthorized accesses to restricted addressable objects. This threat is specific to the additional functionality and does not undermine the SPD defined in the core PP.

- The package introduces the objective of enforcing restricted access to addressable objects, including memory areas. This objective is specific to the additional functionality and does not contradict the objectives defined in the core PP.

- The SFRs fulfil this objective by imposing address-based access controls. These SFRs align with and do not contradicting the SFRs of the core PP.

3 Security Problem Definition

3.1 Assets

56 The assets (related to the core functionality) to be protected are

- the user data of the Composite TOE,

- the Security IC Embedded Software, stored and in operation,
- the security services provided by the TOE for the Security IC Embedded Software.

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

SC1 integrity of user data of the Composite TOE,

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

SC3 correct operation of the security services provided by the TOE for the Security IC Embedded Software.

Note the Security IC Embedded Software is user data and shall be protected while being executed/processed and while being stored in the TOE's protected memories.

58 The Security IC may not distinguish between user data which is public knowledge or confidential. Therefore, the Security IC shall protect the user data of the Composite TOE in integrity and in confidentiality if stored in protected memory areas, unless the Security IC Embedded Software chooses to disclose or modify it.

59 Integrity of the Security IC Embedded Software means that it is correctly being executed which includes the correct operation of the TOE's functionality. Parts of the Security IC Embedded Software which do not contain secret data or security critical source code, may not require protection from being disclosed. Other parts of the Security IC Embedded Software may need to be kept confidential since specific implementation details may assist an attacker.

60 This PP requires the TOE to provide at least one security service: the generation of random numbers by means of a physical Random Number Generator.

61 Section 7 defines the optional functional packages for additional security services. The ST may address these and/or other security services. It is essential that the TOE ensures the correct operation of all the security services provided by the TOE for the Security IC Embedded Software.

62 According to this PP there is the following high-level security concern related to the random number generation service:

SC4 deficiency of random numbers.

63 To be able to protect the assets (SC1 to SC4), the TOE shall self-protect its TSF. Critical information about the TSF shall be protected by the development environment and the operational environment. Critical information may include:

- logical design data, physical design data, IC Dedicated Software, and Configuration Data,
- Initialisation Data and Pre-personalisation Data, specific development aids, test and characterisation related data, material for software development support, and

photomasks.

64 Such information and the ability to perform manipulations assist in threatening the above assets.

65 Note that there are many ways to manipulate or disclose the user data of the Composite TOE: (i) An attacker may manipulate the Security IC Embedded Software or the TOE. (ii) An attacker may cause malfunctions of the TOE or abuse Test Features provided by the TOE. Such attacks usually require design information of the TOE to be obtained. They pertain to all information about (i) the circuitry of the IC (hardware including the physical memories), (ii) the IC Dedicated Software with the parts IC Dedicated Test Software (if any) and IC Dedicated Support Software (if any), and (iii) the TSF Configuration Data. The knowledge of this information may enable or support attacks on the assets. Therefore, the TOE Manufacturer shall ensure that the development and production environment of the TOE (refer to 1.3.5) is secure so that no restricted, sensitive, critical or very critical information is unintentionally made available for attacks in the operational phase of the TOE (cf. [10] for details on assessment of knowledge of the TOE in the vulnerability analysis).

66 The TOE Manufacturer shall apply protection to support the security of the TOE. This not only pertains to the TOE but also to all information and material exchanged with the developer of the Security IC Embedded Software. This covers the Security IC Embedded Software itself if provided by the developer of the Security IC Embedded Software or any authentication data required to enable the download of software. This includes the delivery (exchange) procedures for Phase 1 and the Phases after TOE Delivery as far as they can be controlled by the TOE Manufacturer. These aspects enforce the usage of the supporting documents and the refinements of SAR defined in this protection profile.

67 The information and material produced and/or processed by the TOE Manufacturer in the TOE development and production environment (Phases 2 up to TOE Delivery) can be grouped as follows:

- logical design data,
- physical design data,
- IC Dedicated Software, Initialisation Data and Pre-personalisation Data,
- Security IC Embedded Software, provided by the Security IC Embedded Software developer and implemented by the IC manufacturer,
- specific development aids,
- test and characterisation related data,
- material for software development support, and
- photomasks and products in any form

if they are generated, stored, or processed by the TOE Manufacturer. Explanations can be found in 11.1.3.

3.2 Threats

3.2.1 General

68 The following explanations help to understand the focus of the threats and objectives defined below. For example, certain attacks are only one step towards a disclosure of assets, others may directly lead to a compromise of the application security.

- Manipulation of user data (which includes user data and code of the Composite TOE, stored in or processed by the Security IC) means that an attacker can alter a meaningful block of data. This should be considered for the threats T.Malfunction, T.Phys-Manipulation and T.Abuse-Func.
- Disclosure of user data (which may include user data and code of the Composite TOE, stored in protected memory areas or processed by the Security IC) or TSF data means that an attacker is realistically[2] able to determine a meaningful block of data. This should be considered for the threats T.Leak-Inherent, T.Phys-Probing, T.Leak-Forced and T.Abuse-Func.
- Manipulation of the TSF or TSF data means that an attacker can deliberately deactivate or otherwise change the behaviour of a specific security functionality in a manner which enables exploitation. This should be considered for the threat T.Malfunction, T.Phys-Manipulation and T.Abuse-Func.

69 The cloning of the functional behaviour of the Security IC on its physical and command interface is the highest-level security concern in the application context.

70 The cloning of that functional behaviour requires to (i) develop a functional equivalent of the Security IC Embedded Software, (ii) disclose, interpret and employ the user data of the Composite TOE stored in the TOE, and (iii) develop and build a functional equivalent of the Security IC using the input from the previous steps.

71 The Security IC is a platform for the Security IC Embedded Software which ensures that especially the critical user data of the Composite TOE are stored and processed in a secure way. The Security IC Embedded Software is assumed to ensure that critical user data of the Composite TOE are treated as required in the application context (refer to 3.4). In addition, the personalisation process supported by the Security IC Embedded Software (and perhaps by the Security IC in addition) is assumed secure (refer to 3.4. This last step is beyond the scope of this PP. As a result, the threat “cloning of the functional behaviour of the Security IC on its physical and command interface” is averted by the combination of mechanisms which split into those being evaluated according to this PP (Security IC) and those being subject to the evaluation of the Security IC Embedded Software or Security IC and the corresponding personalisation process. Therefore, functional cloning is indirectly covered by the security concerns and threats described below.

72 The high-level security concerns are refined below by defining threats as required by the CC (refer to Figure 4). Note that manipulation of the TOE is only a means to threaten user data and is not itself a success for the attacker.

[2] Considering the assumed attack potential and, for instance, the probability of errors.

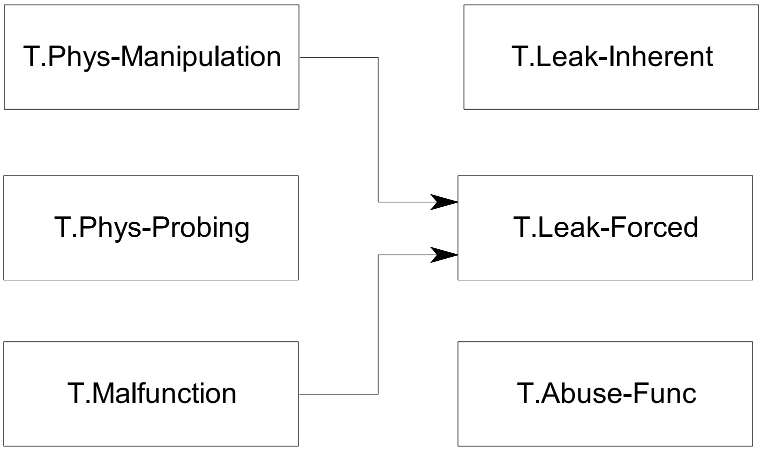


Figure 4: Generic threats

73 The high-level security concern related to security service is refined below by defining threats as required by the CC (refer to Figure 5).

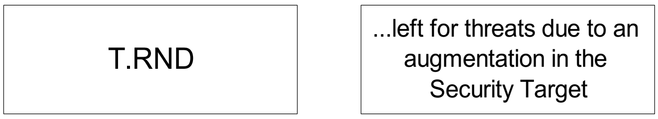

T.RND
...left for threats due to an augmentation in the Security Target

Figure 5: Threats related to security services

Application Note 4: Additional threats may arise if the TOE provides further security functions or security services to the Security IC Embedded Software, The ST author shall complete definition of the threats TOE if necessary.

74 The Security IC Embedded Software may be required to contribute to averting the threats. At least it shall not undermine the security provided by the TOE. For details refer to the assumptions regarding the Security IC Embedded Software specified in 3.4.

75 The security concerns defined in 3.1 are derived from the operational usage by the End-consumer in Phase 7. Indeed,

- Phase 1 and Phases from TOE Delivery up to the end of Phase 6 are covered by assumptions and

- the development and production environments involved in Phase 2 up to TOE Delivery are covered by assurance requirements.

76 The TOE's countermeasures are designed to avert the threats described below. Nevertheless, they may be effective in earlier phases (Phases 4 to 6, refer to Figure 2).

77 The TOE is exposed to different types of influences or interactions with its outer world. Some of them may result from using the TOE only but others may also indicate an attack. The different types of influences or interactions are visualised in Figure 6. Due to the intended use of the TOE, all interactions are assumed to be possible.

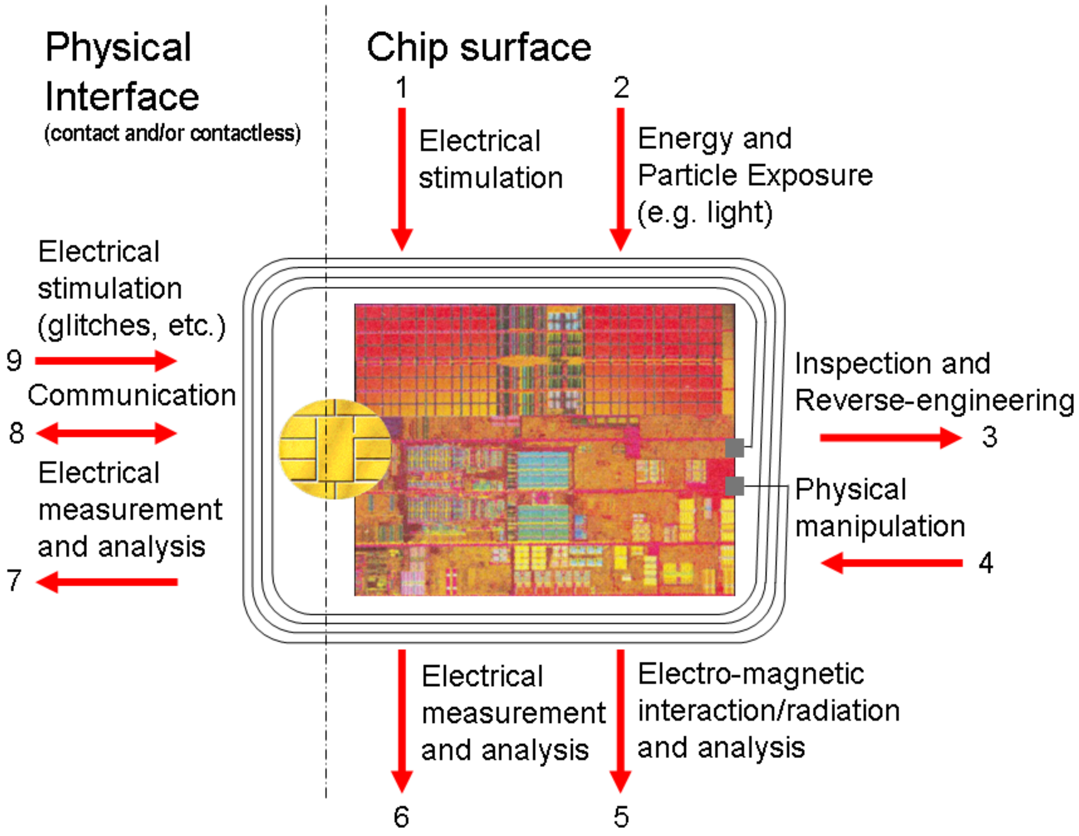

Physical Interface
(contact and/or contactless)
Chip surface
1
Electrical stimulation
2
Energy and Particle Exposure (e.g. light)
Electrical stimulation (glitches, etc.)
9
Communication
8
Electrical measurement and analysis
7
Inspection and Reverse-engineering
3
Physical manipulation
4
Electrical measurement and analysis
6
Electro-magnetic interaction/radiation and analysis
5

Figure 6: Interactions between the TOE and its outer world

78 An interaction with the TOE can be done through the physical interfaces (Number 7 – 9 in Figure 6) which are realised using contacts and/or a contactless interface. Influences or interactions with the TOE also occurs through the chip surface (Number 1 – 6 in Figure 6). In Number 1 and 6 galvanic contacts are used. In Number 2 and 5 the influence (arrow directed to the chip) or the measurement (arrow starts from the chip) does not require a contact. Number 3 and 4 refer to specific situations where the TOE and its functional behaviour are not only influenced but permanent changes are made by applying mechanical, chemical and other methods (such as 1, 2). Many attacks require a prior inspection and some form of reverse-engineering (Number 3). This demonstrates the basic building blocks of attacks. A practical attack will use a combination of these elements.

79 Examples of specific attacks are given in 11.3.

3.2.2 Generic Threats

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

T.Leak-Inherent Inherent Information Leakage

An attacker may exploit information which is leaked from the

TOE during usage of the Security IC in order to disclose confidential user data as part of the assets.

No direct contact with the Security IC internals is required here. Leakage may occur through emanations, variations in power consumption, I/O characteristics, clock frequency, or by changes in processing time requirements. One example is Differential Power Analysis (DPA). This leakage may be interpreted as a covert channel transmission but is more closely related to measurement of operating parameters, which may be derived either from direct (contact) measurements (Numbers 6 and 7 in Figure 6) or measurement of emanations (Number 5 in Figure 6) and can then be related to the specific operation being performed.

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

T.Phys-Probing Physical Probing

An attacker may perform physical probing of the TOE in order (i) to disclose user data while stored in protected memory areas, (ii) to disclose/reconstruct the user data while processed or (iii) to disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.

Physical probing requires direct interaction with the Security IC internals (Numbers 5 and 6 in Figure 6). Techniques commonly employed in IC failure analysis and IC reverse engineering efforts may be used. Before that hardware security mechanisms and layout characteristics need to be identified (Number 3 in Figure 6). Determination of software design including treatment of user data of the Composite TOE may also be a pre-requisite.

This pertains to “measurements” using galvanic contacts or any type of charge interaction whereas manipulations are considered under the threat “Physical Manipulation (T.Phys-Manipulation)”. The threats “Inherent Information Leakage (T.Leak-Inherent)” and “Forced Information Leakage (T.Leak-Forced)“ may use physical probing but require complex signal processing in addition.

82 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 the Security IC Embedded Software by applying environmental stress in order to (i) modify security services of the TOE or (ii) modify functions of the Security IC Embedded Software (iii) deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.

This may be achieved by operating the Security IC outside the normal operating conditions (Numbers 1, 2 and 9 in Figure 6).

The modification of security services of the TOE may e.g. affect the quality of random numbers provided by the random number generator up to undetected deactivation when the random number generator does not produce random numbers and the Security IC Embedded Software gets constant values. In another case errors are introduced in executing the Security IC Embedded Software. To exploit this an attacker needs information about the functional operation, e.g. to introduce a temporary failure within a register used by the Security IC Embedded Software with light or a power glitch.

83 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 Security IC in order to (i) modify user data of the Composite TOE, (ii) modify the Security IC Embedded Software, (iii) modify or deactivate security services of the TOE, or (iv) modify security mechanisms of the TOE to enable attacks disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.

The modification may be achieved through techniques commonly employed in IC failure analysis (Numbers 1, 2 and 4 in Figure 6) and IC reverse engineering efforts (Number 3 in Figure 6). The modification may result in the deactivation of a security feature. Before that hardware security mechanisms and layout characteristics need to be identified. Determination of software design including treatment of user data of the Composite TOE may also be a pre-requisite. Changes of circuitry or data can be permanent or temporary.

In contrast to malfunctions (refer to T.Malfunction) the attacker requires to gather significant knowledge about the TOE’s internal construction here (Number 3 in Figure 6).

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

T.Leak-Forced Forced Information Leakage

An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential user data of the Composite TOE as part of the assets even if the information leakage is not inherent but caused by the attacker.

This threat pertains to attacks where methods described in “Malfunction due to Environmental Stress” (refer to T.Malfunction) and/or “Physical Manipulation” (refer to T.Phys-Manipulation) are used to cause leakage from signals (Numbers 5, 6, 7 and 8 in Figure 6) which normally do not contain significant information about secrets.

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

T.Abuse-Func Abuse of Functionality

An attacker may use functions of the TOE which may not be used after TOE Delivery in order to (i) disclose or manipulate user data of the Composite TOE, (ii) manipulate (explore, bypass, deactivate or change) security services of the TOE or (iii) manipulate (explore, bypass, deactivate or change) functions of the Security IC Embedded Software or (iv) enable an attack disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.

3.2.3 Threats Related to Security Services

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

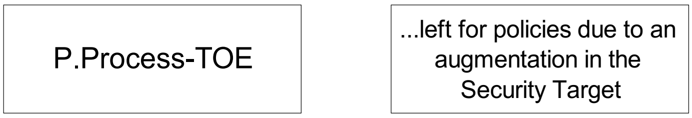
T.RND Deficiency of Random Numbers

An attacker may 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.

An attacker may gather information about the random numbers produced by the TOE security service. Because unpredictability is the main property of random numbers this may be a problem if they are used to generate cryptographic keys. The entropy provided by the random numbers shall be appropriate for the strength of the cryptographic algorithm, the key or the cryptographic variable is 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.

3.3 Organisational Security Policies

87 Figure 7 shows the policies applied in this PP.



P.Process-TOE

...left for policies due to an augmentation in the Security Target

Figure 7: Policies

Application Note 5: Additional OSPs may apply if the TOE provides further security functions or security services which can be used by the Security IC Embedded Software. The ST author shall complete definition of the OSPs if necessary.

88 The IC Developer / Manufacturer shall apply the policy “Identification during TOE Development and Production (P.Process-TOE)” as specified below.

P.Process-TOE Identification during TOE Development and Production

An accurate identification is established for the TOE. This requires that each instantiation of the TOE carries this unique identification.

89 The accurate TOE identification is introduced at the end of the production test in Phase 3. Therefore, the production environment shall support this unique identification.

3.4 Assumptions

90 Figure 8 shows the assumptions applied in this PP.

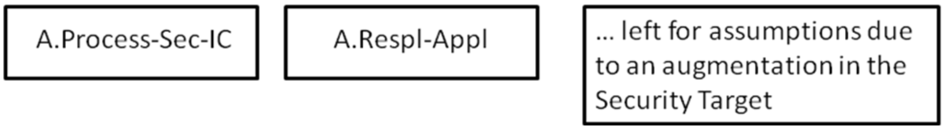

A.Process-Sec-IC
A.Respl-Appl
... left for assumptions due to an augmentation in the Security Target

Figure 8: Assumptions

Application Note 6: Additional assumptions may apply if the TOE provides additional security functions or security services to the Security IC Embedded Software. The ST author shall complete definition of the assumptions, if necessary, cf. [5] ASE_CCL.1.9C.

91 The intended usage of the TOE is twofold, depending on the life cycle phase: (i) The Security IC Embedded Software developer uses it as a platform for the Security IC software being developed. The Composite Product Manufacturer (and the consumer) uses it as a part of the Security IC. The Composite Product is used in a terminal which supplies the Security IC (with power and clock) and (at least) mediates the communication with the Security IC Embedded Software.

92 Before being delivered to the consumer the TOE is packaged. Many attacks require the TOE to be removed from the carrier. Though this extra step adds difficulties for the attacker no specific assumptions are made here regarding the package.

93 Appropriate “Protection during Packaging, Finishing and Personalisation (A.Process-Sec-IC)” shall be ensured after TOE Delivery up to the end of Phase 6, as well as during the delivery to Phase 7 as specified below.

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

It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the End-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).

This means that the phases after TOE Delivery (refer to 1.3.5 and 11.1) are assumed to be protected appropriately. For a preliminary list of assets to be protected refer to paragraph 94 (page 27).

94 The information and material produced and/or processed by the Security IC Embedded Software Developer in Phase 1 and by the Composite Product Manufacturer can be

grouped as follows:

- the Security IC Embedded Software including specifications, implementation and related documentation,
- Pre-personalisation Data and Personalisation Data including the specification of formats and memory areas, test related data,
- the user data of the Composite TOE and related documentation, and
- material for software development support

if they are not under the control of the TOE Manufacturer. Details can be provided in PPs and STs for the evaluation of the Security IC Embedded Software and/or Security IC.

95 The developer of the Security IC Embedded Software must ensure the appropriate usage of Security IC while developing this software in Phase 1 as described in the (i) TOE guidance documents (refer to the CC assurance class AGD) such as the hardware data sheet, and the hardware application notes, and (ii) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as documented in the certification report.

96 Note that specific requirements for the Security IC Embedded Software may not be clear before considering a specific attack scenario during vulnerability analysis of the Security IC (AVA_VAN). A summary of the results is provided in the ETR for composite evaluation (ETR-COMP), cf. [12]. This document is an input for the evaluation of the Composite TOE, cf. [11]. The ETR-COMP may also include guidance for additional tests being required for the combination of hardware and software. The TOE evaluation must be completed before evaluation of the Security IC Embedded Software can be completed. The TOE evaluation can be conducted before and independently from the evaluation of the Security IC Embedded Software.

97 The Security IC Embedded Software shall ensure the appropriate “Treatment of user data of the Composite TOE (A.Resp-Appl)” as specified below.

A.Resp-Appl Treatment of user data of the Composite TOE

All user data of the Composite TOE are under the control of the Security IC Embedded Software. Therefore, it is assumed that security relevant user data of the Composite TOE (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.

The application context specifies how the user data of the Composite TOE shall be handled and protected. The evaluation of the Security IC according to this PP is conducted on a generalized application context. The concrete requirements for the Security IC Embedded Software shall be defined in the PP or ST for the Security IC Embedded Software. The Security IC cannot prevent any compromise or modification of user data of the Composite TOE by malicious Security IC Embedded Software.

4 Security Objectives

4.1 Security Objectives for the TOE

4.1.1 General

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

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

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

SG3 maintain the correct operation of the security services provided by the TOE for the Security IC Embedded Software.

Note, the Security IC may not distinguish between user data which are publicly known or kept confidential. Therefore, the Security IC shall protect the user data in integrity and in confidentiality if stored in protected memory areas, unless the Security IC Embedded Software chooses to disclose or modify it. Parts of the Security IC Embedded Software which do not contain secret data or security critical source code, may not require protection from being disclosed. Other parts of the Security IC Embedded Software may need kept confidential since specific implementation details may assist an attacker.

99 These standard high-level security goals in the context of the security problem definition constitute the starting point for the definition of security objectives as required by the CC (refer to Figure 9). Note that the integrity of the TOE is a means to reach these objectives.

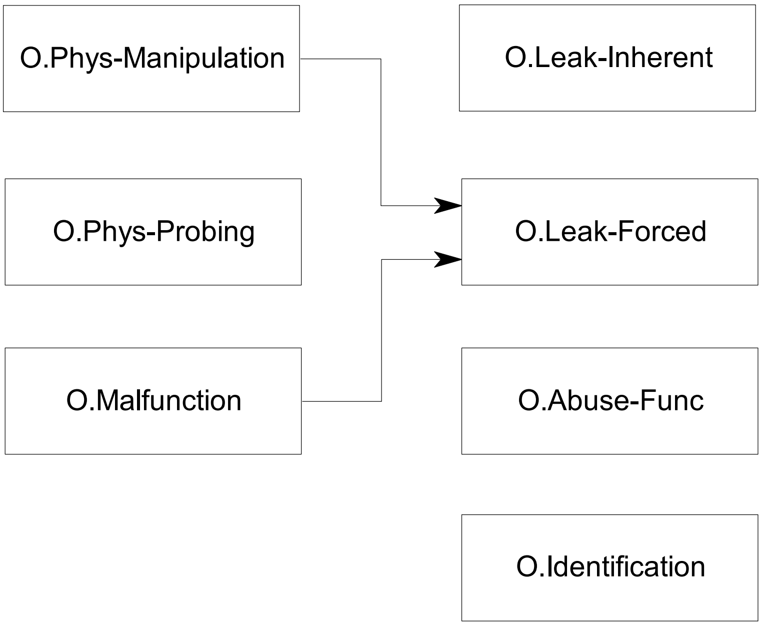


Figure 9: Generic security objectives

100 According to this PP the following high-level security goal related to specific functionality holds:

SG4 provide true random numbers.

101 The additional high-level security considerations are refined below by defining security objectives as required by the CC (refer to Figure 10).

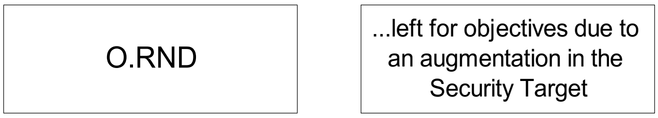

O.RND
...left for objectives due to an augmentation in the Security Target

Figure 10: Security objectives related to specific functionality

Application Note 7: Additional objectives may arise if the TOE provides further security functions or security services to the Security IC Embedded Software. The ST author shall complete definition of the objectives if necessary.

4.1.2 Generic security objectives for the TOE

102 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 data stored and/or processed in the Security IC

- by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines) and
- by measurement and analysis of the time between events found by measuring signals (for instance on the power, clock, or I/O lines).

103 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 is not given here.

104 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/reconstruction of user data while stored in protected memory areas and processed or against the disclosure of other critical information about the operation of the TOE.

This includes protection against

- measuring through galvanic contacts which is direct physical probing on the chips surface except on pads being bonded (using standard tools for measuring voltage and current) or

- measuring not using galvanic 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.

105 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. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields.

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 (refer to the objective O.Phys-Manipulation) provided that detailed knowledge about the TOE´s internal construction is required, and the attack is performed in a controlled manner.

106 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 (including its software and TSF data), the Security IC Embedded Software and the user data of the Composite TOE. This includes protection against

- reverse-engineering (understanding the design and its properties and functions),

- manipulation of the hardware and any data, as well as

- undetected manipulation of memory contents.

107 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.

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

O.Leak-Forced Protection against Forced Information Leakage

The TOE shall be protected against disclosure of confidential data processed in the Security IC (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker

- by forcing a malfunction (refer to "Protection against Malfunction due to Environmental Stress (O.Malfunction)" and/or
- by a physical manipulation (refer to "Protection against Physical Manipulation (O.Phys-Manipulation)".

If this is not the case, signals which normally do not contain significant information about secrets could become an information channel for a leakage attack.

109 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 that functions of the TOE which may not be used after TOE Delivery can be abused in order to (i) disclose critical user data of the Composite TOE, (ii) manipulate critical user data of the Composite TOE, (iii) manipulate Security IC Embedded Software or (iv) bypass, deactivate, change or explore security features or security services of the TOE.

110 Details depend, for instance, on the capabilities of the Test Features provided by the IC Dedicated Test Software, which are not specified in this PP.

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

O.Identification TOE Identification

The TOE shall provide means to store Initialisation Data and Pre-personalisation Data in its non-volatile memory. The Initialisation Data (or parts of them) are used for TOE identification.

4.1.3 Security objectives related to the TOE security services

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

O.RND Random Numbers

The TOE shall ensure the cryptographic quality of random number generation. For instance, random numbers shall not be predictable and shall have a sufficient entropy.

The TOE shall ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.

4.2 Security Objectives for the Security IC Embedded Software

113 The development of the Security IC Embedded Software is outside the development and manufacturing of the TOE (cf. 1.3.5). The Security IC Embedded Software defines the operational use of the TOE. This section describes the security objectives for the Security IC Embedded Software.

114 In order to ensure that the TOE is used in a secure manner, the Security IC Embedded 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 IC Dedicated Software of the TOE, (iii) TOE application notes, other guidance documents, and (iv) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as referenced in the certification report.

115 The Security IC Embedded Software shall provide “Treatment of user data of the Composite TOE (OE.Resp-Appl)” as specified below.

OE.Resp-Appl Treatment of user data of the Composite TOE

Security relevant user data of the Composite TOE (especially cryptographic keys) shall be treated by the Security IC Embedded Software as required by the security needs of the specific application context.

For example, the Security IC Embedded Software will not disclose security relevant user data of the Composite TOE to unauthorised users or processes when communicating with a terminal.

4.3 Security Objectives for the Operational Environment

116 Appropriate “Protection during Packaging, Finishing and Personalisation (OE.Process-Sec-IC)” shall be ensured after TOE Delivery up to the end of Phases 6, as well as during the delivery to Phase 7 as specified below.

OE.Process-Sec-IC Protection during composite product manufacturing

Security procedures shall be used after TOE Delivery up to delivery to the End-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).

This means that Phases after TOE Delivery up to the end of Phase 6 (refer to 1.3.5) shall be protected appropriately. For a preliminary list of assets to be protected refer to paragraph 94 (page 27).

4.4 Security Objectives Rationale

117 Table 4-1 gives an overview of how assumptions, threats, and organisational security policies are addressed by the security objectives.

Table 4-1: Security Objectives versus Assumptions, Threats and Policies

Assumption, Threat or OSP	Security Objective	Notes
A.Resp-Appl	OE.Resp-Appl	
P.Process-TOE	O.Identification	Phases 2 – 3 optional Phase 4
A.Process-Sec-IC	OE.Process-Sec-IC	Phases 5 – 6 optional Phase 4
T.Leak-Inherent	O.Leak-Inherent	
T.Phys-Probing	O.Phys-Probing	
T.Malfunction	O.Malfunction	
T.Phys-Manipulation	O.Phys-Manipulation	
T.Leak-Forced	O.Leak-Forced	
T.Abuse-Func	O.Abuse-Func	
T.RND	O.RND	

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

119 Since OE.Resp-Appl requires the Security IC Embedded Software to implement measures as assumed in A.Resp-Appl, the assumption is covered by the objective.

120 The justification related to the organisational security policy “Protection during TOE Development and Production (P.Process-TOE)” is as follows:

121 O.Identification requires that the TOE supports the possibility of a unique identification. The unique identification can be stored on the TOE. Since the unique identification is generated by the production environment, the production environment shall support the integrity of the generated unique identification. The technical and organisational security controls that ensure the security of the development environment and production environment are evaluated based on the assurance controls that are part of the evaluation. For a list of material produced and processed by the TOE Manufacturer refer to paragraph 67 (page 20). All listed items and the associated development and

production environments are subject of the evaluation. Therefore, the organisational security policy P.Process-TOE is covered by this objective, as far as organisational controls are concerned.

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

123 Since OE.Process-Sec-IC requires the Composite Product Manufacturer to implement those measures assumed in A.Process-Sec-IC, the assumption is covered by this objective.

124 The justification related to the threats “Inherent Information Leakage (T.Leak-Inherent)”, “Physical Probing (T.Phys-Probing)”, “Malfunction due to Environmental Stress (T.Malfunction)”, “Physical Manipulation (T.Phys-Manipulation)”, “Forced Information Leakage (T.Leak-Forced)“, “Abuse of Functionality (T.Abuse-Func)” and “Deficiency of Random Numbers (T.RND)” is as follows:

125 For all threats the corresponding objectives (refer to Table 4-1) are stated in a way, which directly corresponds to the description of the threat (refer to 3.2). It is clear from the description of each objective (refer to 4.1), that the corresponding threat is removed if the objective is valid. More specifically, in every case the ability to use the attack method successfully is countered, if the objective holds.

5 Extended Components Definition

5.1 Definition of the Family FAU_SAS

126 To define the security functional requirements of the TOE an additional family (FAU_SAS) of the Class FAU (Security Audit) is defined here. This family describes 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.

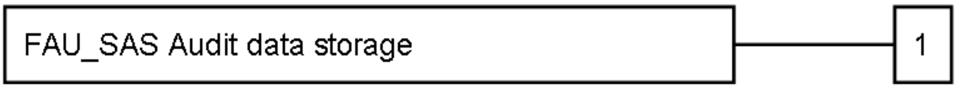
127 The family “Audit data storage (FAU_SAS)” is specified as follows.

FAU_SAS Audit data storage

Family behaviour

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

Component levelling



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

Management: FAU_SAS.1

There are no management activities foreseen.

Audit: FAU_SAS.1

There are no actions defined to be auditable.

FAU_SAS.1 **Audit storage**

Hierarchical to: No other components.

Dependencies: No dependencies.

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

6 Security Requirements

6.1 Security Functional Requirements for the TOE

6.1.1 General

128 Figure 11 shows the generic security functional requirements.

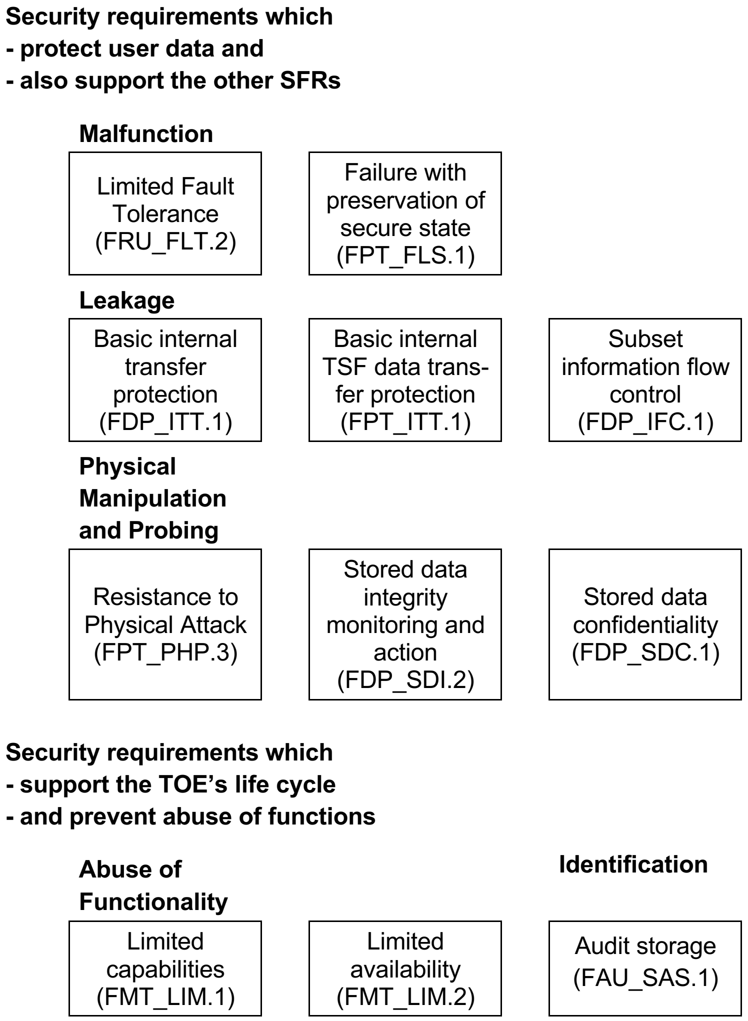

Security requirements which
- protect user data and
- also support the other SFRs
Malfunction
Limited Fault Tolerance (FRU_FLT.2)
Failure with preservation of secure state (FPT_FLS.1)
Leakage
Basic internal transfer protection (FDP_ITT.1)
Basic internal TSF data transfer protection (FPT_ITT.1)
Subset information flow control (FDP_IFC.1)
Physical Manipulation and Probing
Resistance to Physical Attack (FPT_PHP.3)
Stored data integrity monitoring and action (FDP_SDI.2)
Stored data confidentiality (FDP_SDC.1)
Security requirements which
- support the TOE's life cycle
- and prevent abuse of functions
Abuse of Functionality
Identification
Limited capabilities (FMT_LIM.1)
Limited availability (FMT_LIM.2)
Audit storage (FAU_SAS.1)

Figure 11: Generic SFRs

129 Figure 12 shows the security functional requirements related to the security services included in the core PP.

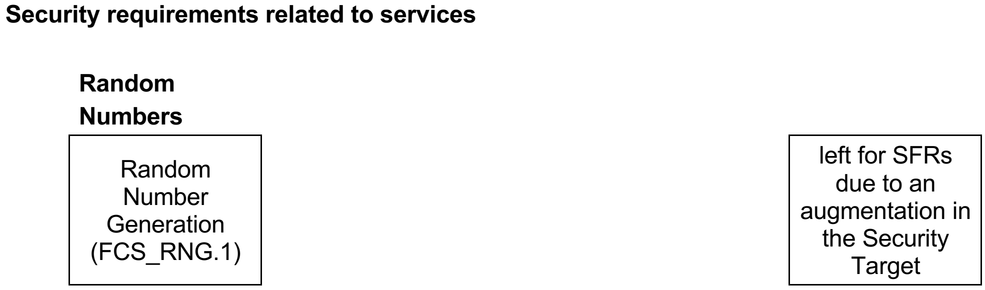

Security requirements related to services
Random Numbers
Random Number Generation (FCS_RNG.1)
left for SFRs due to an augmentation in the Security Target

Figure 12: SFRs related to security services

Application Note 8: Additional SFRs may be required if the TOE provides further security functions or security to the Security IC Embedded Software. The ST author shall complete definition of the SFRs if necessary, 130 In order to define the SFRs, Part 2 of the CC was used. However, some SFRs have been refined. The refinements are described below the associated SFR.

131 The refinement operation is used to add details to a requirement, and, thus, further restricts it. Refinements of SFRs 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 a case, an extra paragraph labelled "Refinement" may be given.

132 The selection operation is used to select one or more options provided by the CC for specific requirements. In this PP, selections 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.

133 The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. In this PP, assignments are denoted by 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 in the PP defines a selection to be performed by the ST author. Thus, this text is underlined and italicised like this.

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

135 In this PP operations are completed for all security functional components except the components FCS_RNG.1 (Generation of random numbers), FAU_SAS.1 (Audit storage), FDP_SDC.1 (Memory protection) and FDP_SDI.2 (Stored data integrity monitoring and action). Some of the components defined in the functional packages are also not completed. The ST author shall perform all the operations that are left open in this PP.

6.1.2 Malfunction

136 There are different ranges of operating conditions such as supply voltage, external frequency and temperature. The TOE can be operated within the limits visualised as the inner dashed rounded rectangle in Figure 13 and shall operate correctly there. The limits have been reduced to ensure correct operation. This is visualised by the outer dotted rounded rectangle in the figure.

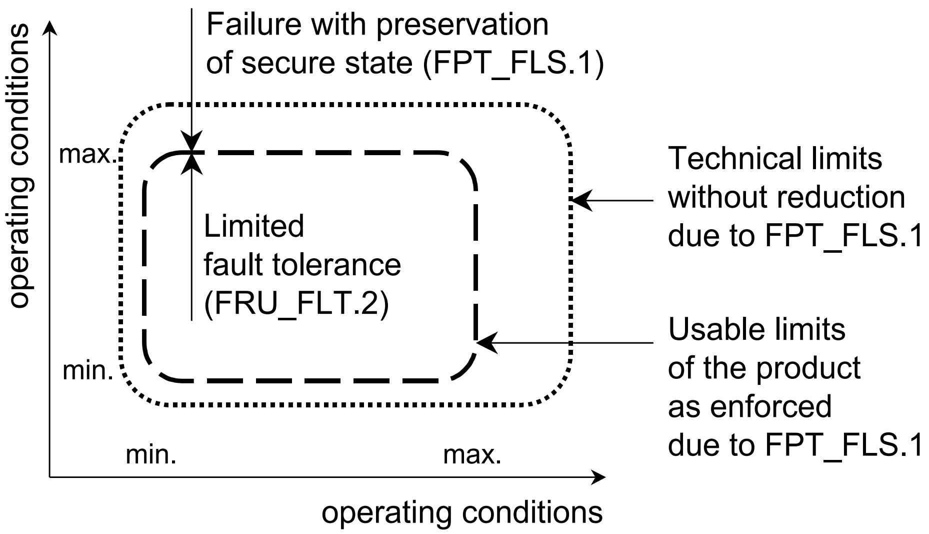

operating conditions
Failure with preservation of secure state (FPT_FLS.1)
max.
min.
Limited fault tolerance (FRU_FLT.2)
Technical limits without reduction due to FPT_FLS.1
Usable limits of the product as enforced due to FPT_FLS.1
min.
max.
operating conditions

Figure 13: Paradigm regarding Operating Conditions

137 Figure 13 shall not be understood as being two-dimensional and defining static limits only. Reality is multi-dimensional and includes a variety of timing aspects. Note that the limit of the operating conditions visualised by the inner dashed rounded rectangle in Figure 13 is not necessarily exactly reflected by the limits identified in the TOE's data sheet. Instead, this limit marks the boundary between the “tolerance reaction” of the TOE and the “active reaction” of sensors (and perhaps other circuitry).

138 The security functional component FRU_FLT.2 has been selected to address the robustness within some limit (as shown by the inner dashed rectangle in Figure 13) before active reaction takes place to reach a failure with preservation of secure state. Note that in most cases the TOE does not (detect faults or failures and then correct them to guarantee further operation of all the TOE's capabilities. This is the way software would implement Limited fault tolerance (FRU_FLT.2). Instead, the TOE will achieve the same by (i) providing a stable functional design within the limits of operational conditions (e.g. temperature) and (ii) eliminating the cause for possible faults and by being resistant against influences (e.g. robustness against glitches of the power supply by means of filtering). In the case of the TOE the “reaction to a failure” is replaced by the “reaction to operating conditions” which could cause a malfunction without the reaction of the TOE's countermeasure addressed by the security functional component FPT_FLS.1.

139 If the TOE is exposed to other operating conditions this may not be tolerated. Then the TOE shall detect that and preserve a secure state (use of detectors and cause a reset for instance). The security functional component FPT_FLS.1 has been selected to ensure that. The way the secure state is reached depends on the implementation. Note that the TOE can monitor both external operating conditions and other internal conditions and then react appropriately. Exposure to specific “out of range” external operating conditions (environmental stress) may cause internal failure conditions which cannot be tolerated by FRU_FLT.2. Referring to external operating conditions the TOE is expected to respond if conditions are detected which may cause a failure. Examples for implementations of the security functional requirement FPT_FLS.1 are a voltage detector (external condition) and a circuitry which detects accesses to address areas which are not used (internal condition).

140 Those parts of the TOE which support the security functional requirements “Failure with preservation of secure state (FPT_FLS.1)” and “Limited Fault tolerance (FRU_FLT.2)” shall be protected from misconfiguration of and by-passing by means of the Security IC Embedded Software. These aspects are addressed by the security architecture assurance requirements (ADV_ARC.1).

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

FRU_FLT.2 **Limited fault tolerance**

Hierarchical to: FRU_FLT.1 Degraded fault tolerance

Dependencies: FPT_FLS.1 Failure with preservation of secure state.

FRU_FLT.2.1 The TSF shall ensure the operation of all the TOE’s capabilities when the following failures occur: exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1)[3].

Refinement: **The term “failure” above means “circumstances”. The TOE prevents failures for the “circumstances” defined above.**

Application Note 9: Environmental conditions include but are not limited to power supply, clock, and other external signals (e.g. reset signal) necessary for the TOE operation.

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

FPT_FLS.1 **Failure with preservation of secure state**

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: exposure to operating conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2) and where therefore a malfunction could occur[4].

Refinement: **The term “failure” above also covers “circumstances”. The TOE prevents failures for the “circumstances” defined above.**

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

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

Application Note 10: The ST shall describe the secure state. The ST author should provide a clear definition of the secure state and the rationale supporting that definition.

Application Note 11: The CC suggest that the TOE generates audit data for the security functional requirements 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 ST should consider this especially for FPT_FLS.1.

6.1.3 Abuse of Functionality

143 During testing at the end of Phase 3, before TOE Delivery, the TOE shall be able to store some data (for instance about the production history or identification data of the individual die or other data to be used after delivery). Therefore, the security functional component Audit storage (FAU_SAS.1) has been added. The security functional component FAU_SAS.1 has been newly created (refer to 5.1 and is used instead of FAU_GEN.1 which is too comprehensive to be applicable in this context.

144 The requirement FAU_SAS.1 shall be regarded as covering the injection of Initialisation Data, Pre-personalisation Data or other data as described in 11.1.2. After TOE Delivery the identification data (injected as part of the Initialisation Data) and the Pre-personalisation Data are available to the Security IC Embedded Software. These data are protected by the TOE as all other user data of the Composite TOE. It’s up to the Security IC Embedded Software to use these data stored and provided by the TOE.

145 Each instantiation of the TOE shall undergo exhaustive testing at clearly defined stages of the production process where the correct functioning and properties are ascertained and, if necessary, information may be stored in the NVM. This task is done by a specialised group of people of the TOE Manufacturer called “test-personnel”. The test-personnel is the first user of the TOE, and their identity may be assumed as default user for FAU_SAS.1. If the Initialisation Data, Pre-personalisation Data or assigned other data can be written only once the test-personnel will be the only user able to store these data.

146 The TOE shall prevent functions provided by the IC Dedicated Test Software or by hardware features, called Test Features, from being abused after TOE Delivery to compromise the TOE’s security. This includes but is not limited to disclose or manipulate user data of the Composite TOE, and bypass, deactivate, change or explore security features or functions of the TOE. Details depend on the capabilities of the Test Features provided by the IC Dedicated Test Software and/or the hardware.

147 This can be achieved (i) by limiting the capabilities of the Test Features after Phase 3, (ii) by limiting the availability of these Test Features after Phase 3 or (iii) by a combination of both. The security functional components Limited capabilities (FMT_LIM.1) and Limited availability (FMT_LIM.2) address this. The Limited capability and availability policy applies to both FMT_LIM.1 and FMT_LIM.2.

148 Examples of technical mechanisms used in the TOE are password-based user authentication, non-availability through removal or disabling by fusing, or a combination of both. A detailed technical specification would unnecessarily disclose details and is beyond the scope of a specification of requirements.

149 The TOE is tested after production in Phase 3 (refer to 11.1.2) using means provided by the IC Dedicated Software and/or specific hardware. The IC Dedicated Software is considered a test tool that is delivered as part of the TOE and used before TOE Delivery only. It does not provide functions in later phases of the Security IC's life cycle. Therefore, no security functional requirement is mandatory according to this PP regarding these testing capabilities except FPT_LIM.1 and FPT_LIM.2.

150 All necessary information about the capabilities of the Test Features (including the IC Dedicated Software) shall be provided by TOE design (ADV_TDS) for the description of the mechanisms and the security architecture (ADV_ARC) for the description of the security architecture design and implementation to limit the availability of the Test Features. The Vulnerability Assessment (AVA) shall analyse the effectiveness of the security mechanisms to enforce FMT_LIM.1 and FMT_LIM.2. For further information on how to handle the Test Features refer to 6.2.2.

151 The TOE shall meet the requirement "Limited capabilities (FMT_LIM.1)".

FMT_LIM.1 **Limited capabilities**

Hierarchical to: No other components.

Dependencies: FMT_LIM.2 Limited availability.

FMT_LIM.1.1 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[5].

152 The TOE shall meet the requirement "Limited availability (FMT_LIM.2)".

FMT_LIM.2 **Limited availability**

Hierarchical to: No other components.

Dependencies: FMT_LIM.1 Limited capabilities.

FMT_LIM.2.1 The TSF shall be designed in a manner that limits its availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: 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[6].

153 The TOE shall meet the requirement "Audit storage (FAU_SAS.1)" (CC Part 2 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 [5] with the capability to store [selection: *the Initialisation Data, Pre-personalisation Data,* [assignment: *other data*]] [6] in the [assignment: *type of persistent memory*].

Application Note 12: The integrity and uniqueness of the unique identification of the TOE shall be supported by the development, production and test environments. For details refer to 6.2.2.1.

Application Note 13: The test process is running under control of the test-personnel. The ST author shall perform the operation in the element FAU_SAS.1.1 by assigning the data and the type of persistent memory provided for the storage of Initialisation Data and/or Pre-personalisation Data and/or other data e.g. like supplements of the Security IC Embedded Software. If the TOE provides specific functions to protect these data or to process them, appropriate security functional requirements can be specified in the ST, supported by explanatory text.

6.1.4 Physical Manipulation and Probing

154 The TOE can be subject to “tampering” which here pertains to (i) manipulation of the chip hardware and its security features with (ii) prior reverse-engineering to understanding the design and its properties and functions), (iii) determination of critical data through measuring using galvanic contacts, (iv) determination of critical data not using galvanic contacts and (v) calculated manipulation of memory contents. Refer to paragraph 68 (on page 21) for further explanations.

155 The TSF protects the user data stored in specified memory areas against compromise and undetected manipulation. The TSF provides access to the data in the memory through the specified interfaces only. The TSF protects the information of the user data stored in specified memory areas against compromise by physical access to the stored data bypassing these interfaces. Therefore, the security functional component Stored data confidentiality (FDP_SDC.1) has been selected. Although the TSF may not prevent the manipulation of the memory content, it shall monitor the memory for integrity errors and react on detected errors as required by Stored data integrity monitoring (FDP_SDI.2).

156 The TOE is not always powered and therefore not able to detect, react or notify that it has been subject to tampering. Nevertheless, its design characteristics make reverse-engineering, manipulation, etc. more difficult. This is regarded as being an “automatic response” to tampering. Therefore, the security functional component Resistance to

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

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

physical attack (FPT_PHP.3) has been selected. The TOE may also provide features to actively respond to a possible tampering attack which is also covered by FPT_PHP.3.

157 The TOE may also leave it up to the Security IC Embedded Software to react when a possible tampering has been detected. Comprehensive guidance (refer to CC assurance class AGD) will be given for the developer of the Security IC Embedded Software in this case.

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

FDP_SDC.1 **Stored data confidentiality**

Hierarchical to: No other components.

Dependencies: No dependencies.

FDP_SDC.1.1 The TSF shall ensure the confidentiality of [selection: *all user data, the following user data* [assignment: *list of user data*]] while it is stored in the [selection: *temporary memory, persistent memory, any memory*].

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

FDP_SDI.2 Stored data integrity monitoring and action

Hierarchical to: FDP_SDI.1 Stored data integrity monitoring

Dependencies: No dependencies.

FDP_SDI.2.1 The TSF shall monitor user data stored in containers controlled by the TSF for [assignment: *integrity errors*] on all objects, based on the following attributes: [assignment: *user data attributes*].

FDP_SDI.2.2 Upon detection of a data integrity error, the TSF shall [assignment: *action to be taken*].

Application Note 14: The ST author shall perform the open operations and may assign the monitored memory areas to the user data attributes in the element FDP_SDI.2.1.

160 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[7] to the TSF[8] by responding automatically such that the SFRs are always enforced.

Refinement: **The TSF shall implement mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of physical attacks (especially manipulation) the TSF can by no means detect attacks on all its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, "by responding automatically" means that (i) attacks can occur anytime and (ii) countermeasures are permanently provided.**

Application Note 15: The ST shall describe the automatic response of the TSF. All security functional requirements are derived from security objectives to protect the user data stored and processed on the Security IC or to provide secure security services. Therefore, the security functional requirements are enforced if the TOE stops operating 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.

Application Note 16: The TOE may also provide "unambiguous detection of physical tampering that can compromise the TSF" and "the capability to determine whether physical tampering with the TSF's devices or TSF's elements has occurred" as required by the elements FPT_PHP.1.1 and FPT_PHP.1.2. However, the notification of the tampering is subject to the Security IC Embedded Software. The ST author can mention the security features that support the detection of physical tampering so that the author of a composite ST is able to define an associated security functional requirement.

6.1.5 Leakage

161 When the Security IC processes user data of the Composite TOE and/or TSF Data, information about these data may be leaked by signals which can be measured externally (e.g. through the ISO contacts of a smartcard). An attacker may also cause malfunctions or perform manipulations of the TOE to induce the TOE to leak information. The analysis of those measurement data can lead to the disclosure of user data of the Composite TOE and other critical data. Examples are given in 11.3.

162 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 shall resist leakage attacks both for user data of the Composite TOE and TSF data. The corresponding security policy is defined in the security functional requirement "Subset information flow control (FDP_IFC.1)". These security functional requirements address inherent leakage. With respect to forced leakage they

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

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

are considered in combination with the security functional requirements “Limited fault tolerance (FRU_FLT.2)” and “Failure with preservation of secure state (FPT_FLS.1)” on the one hand and “Resistance to physical attack (FPT_PHP.3)” on the other.

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

FDP_ITT.1 **Basic internal transfer protection**

Hierarchical to: No other components.

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

FDP_ITT.1.1 The TSF shall enforce the Data Processing Policy[9] to prevent the disclosure[10] 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.**

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

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

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure[11] when it is transmitted between separate parts of the TOE.

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

This requirement is equivalent to FDP_ITT.1 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.

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

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

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

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

FDP_IFC.1 **Subset information flow control**

Hierarchical to: No other components.

Dependencies: FDP_IFF.1 Simple security attributes

FDP_IFC.1.1 The TSF shall enforce the Data Processing Policy[12] on all confidential data when they are processed or transferred by the TOE or by the Security IC Embedded Software[13].

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

“User data of the Composite TOE and TSF data shall not be accessible from the TOE except when the Security IC Embedded 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 Security IC Embedded Software.”

6.1.6 Random Numbers

167 The TOE shall meet the requirement “Random number generation (FCS_RNG.1)” as specified below.

FCS_RNG.1 **Random number generation**

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1 The TSF shall provide a [selection: *physical, hybrid physical, hybrid deterministic*][14] 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 17: The ST author 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 Security IC. See 11.2for guidance on the instantiation of FCS_RNG.1.

6.2 Security Assurance Requirements for the TOE

[12] [assignment: information flow control SFP]

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

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

6.2.1 General

168 An ST that is conformant with this PP shall be evaluated against the class ASE.

169 The security assurance requirements for the TOE are those defined in

Evaluation Assurance Level 4 (EAL4)

and augmented by taking the following components:

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

170 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 procedures	(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 18: An ST that is conformant to this PP can claim higher hierarchical components than those defined. To support this, this PP often refers to “the CC assurance component of the family XY” instead of referring to the specific components listed above. If the ST claims further augmentations these shall be identified in this section and in 2.3. The ST author also extend the rationale of this PP as appropriate.

6.2.2 Refinements of the TOE Assurance Requirements

171 The evaluation and certification of smartcards and similar devices are subject to requirements defined in CCRA and EUCC supporting documents such as [6], [9], [10], and [11]. These documents are regularly updated; evaluations shall be conducted based on in their latest versions unless stated otherwise.

172 The refinements of the assurance requirements are related to the specificities of the Security IC development and production process (Security IC’s life cycle), and support the comparability of evaluations conformant to this PP.

173 Where refinements are not needed some background information based on the mandatory supporting documents is provided. In all cases the background information is informative only. The mandatory documents shall be consulted for exact details and overrule the refinements in case of any inconsistency (e.g. due to updates).

174 Refinements are given in **bold**. These refinements refer to the underlined keywords within the security assurance requirements.

Application Note 19: The refinements defined below may also apply to a hierarchically higher assurance component of the specific family. If an ST includes an additional augmentation, the author of the ST shall determine if the refinements as defined below remain applicable.

6.2.2.1 Refinement of Delivery Procedure (ALC_DEL)

Introduction

175 The CC assurance component of the family ALC_DEL (Delivery procedures) refers to the delivery of (i) the TOE or parts of it (ii) to the user or user’s site (Developer of the Security IC Embedded Software or the Composite TOE Manufacturer). The CC 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 other assigned data.

176 In the case of a Security IC, more “material and information” than the TOE itself (which, by definition, includes the necessary guidance) is exchanged with “users”. Therefore, considering the definition of the CC the following refinement is made regarding the items “TOE” and “to the user or user’s site”:

177 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 downstream user.

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 downstream user.

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

178 **For delivery of the TOE to the "Composite Product Manufacturer" as downstream user, all the external interfaces of the TOE Manufacturer shall be considered. These are:**

- **the interface with the Security IC Embedded Software Developer (Phase 1) where information about the Security IC, 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 form of wafers, sawn wafers (dice) or packaged products are exchanged.**

Application Note 20: The downstream user in the context of ALC_DEL is the Composite Product Manufacturer to which the TOE as Security IC is delivered. The End-consumer is the consumer of the Composite Product which includes the TOE as platform for the IC Embedded Software.

Application Note 21: All identified critical information about the TOE is covered to avoid any tampering with the actual version or the substitution of a false version (including unauthorised modification or replacement).

Application Note 22: Depending on whether the TOE comprises programmable NVM and/or ROM, in addition to IC pre-personalisation requirements, the Security IC Embedded Software and/or keys for the authorised personalisation of the programmable NVM are delivered to the Composite Product Manufacturer.

6.2.2.2 Refinements of Development Security (ALC_DVS)

Introduction

179 The document [13] applies.

180 The CC assurance component of the family ALC_DVS refers to (i) the “development environment”, (ii) the “TOE” or “TOE design and implementation”. The component ALC_DVS.2 “Sufficiency of security controls” requires additional evidence for the suitability of the security controls.

181 The TOE Manufacturer shall ensure that the development and production of the TOE (refer to 1.3.5) is secure so that no restricted, sensitive, critical or very critical information is unintentionally made available in the operational phase of the TOE which may enable or support attacks (cf. [10] for details). Therefore, confidentiality and integrity of design information and test data shall be guaranteed, access to samples[15], development tools and other material shall be restricted to authorised persons only, scrap shall be destroyed. This not only pertains to the TOE but also to all information and material exchanged with the developer of the Security IC Embedded Software and therefore especially to the Security IC Embedded Software itself. This includes the delivery (exchange) procedures for Phase 1 and the Phases after TOE Delivery as far as they can be controlled by the TOE Manufacturer.

182 In the case of a Security IC the TOE is developed and produced within a complex industrial process which shall especially be protected. Therefore, the following refinement is made regarding the items “development environment”, “TOE design and implementation” and the confirmation of the application of the security controls:

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

Developer action elements:

ALC_DVS.2.1D The developer shall produce and provide development environment security documentation.

Content and presentation elements:

ALC_DVS.2.1C The development environment security documentation shall describe all the physical, logical, procedural, personnel, and other security controls that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.2.2C The development environment security documentation shall justify that the security controls provide the necessary level of protection to maintain the confidentiality and integrity of the TOE.

Evaluator action elements:

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

ALC_DVS.2.2E The evaluator shall confirm that the security controls are being applied.

[15] This may comprise so called open samples that are only used for evaluation purposes.

Refinement

184 **“TOE design and implementation” comprises all material and information related to the development and production environments of the TOE. Therefore, all critical information identified paragraph 63, shall be considered in order to ensure integrity and – if necessary confidentiality - (including protection against unauthorised disclosure, unauthorised modification or replacement and theft). The “development environment security documentation” shall describe all security controls related to the “TOE design and implementation” in the development environment as defined above.**

Application Note 23: Whenever samples, material and information are given to external partners (such as the developer of the Security IC Embedded Software) the latter shall be obliged by a Non-Disclosure Agreement to treat the samples, material and information as it is required for the TOE Manufacturer.

Background information

185 The scope of the requirement “Development environment security (ALC_DVS)” pertains to the Phase 2 up to TOE Delivery. These phases are under the control of the TOE Manufacturer. The “development environment” as referred to in the CC covers both the TOE development environment (Phase 2) and the TOE production environment (Phase 3 and also Phase 4 if the TOE Manufacturer delivers packaged products).

6.2.2.3 Refinement of CM Scope (ALC_CMS)

Introduction

186 The CC assurance component of the family ALC_CMS (CM scope) refers to the tracking of specific configuration items within the developer’s configuration management system.

187 In the case of a Security IC, it is helpful to clarify the scope of the configuration item “TOE implementation representation”:

188 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

189 **Although the Security IC Embedded Software is user data (not part of the TOE), the whole Security IC Embedded Software or part of it may be delivered together with the TOE, e.g. when implemented in ROM or written by the TOE manufacturer in NVM. Therefore, the Security IC Embedded Software and any authentication data required for loading shall be included in the configuration list only if the TOE Manufacturer has control over these items.**

Background information

190 Since the Security IC Embedded Software may not be developed by the TOE Manufacturer it is only available in a specific form and is not part of the TOE even if it is delivered together with it. Authentication data may be required for products implementing programmable NVM to enable the download of software.

191 Depending on the product type with programmable NVM and/or ROM the Security IC Embedded Software and/or authentication data for a secure loader of the programmable NVM may be considered as part of the TOE implementation representation.

192 The “TOE implementation representation” within the scope of the CM includes at least:

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

6.2.2.4 Refinement of CM Capabilities (ALC_CMC)

Introduction

193 The CC 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.

194 In the case of a Security IC, the scope of “configuration items” and the meaning of “TOE” need to be clarified:

195 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 unique 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 or methods 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 controls 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

196 **“Configuration items” comprise all items defined and refined under ALC_CMS, which shall be tracked under CM.**

197 **A production control system shall be applied to guarantee the traceability and completeness of different production charges or lots. The number of wafers, dies and chips shall be tracked by this system. Appropriate administration**

procedures shall 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 shall be ensured that these wafers, dies or assembled devices are returned to the same production stage from which they are taken, or they shall be securely stored or destroyed otherwise.

6.2.2.5 Refinement of Security Architecture (ADV_ARC)

Introduction

198 The “Security architecture requirements (ADV_ARC) for smart cards and similar devices” [9] provides guidance on how to apply the assurance requirements for the security architecture to security integrated circuits.

199 The refinement of the CC assurance component ADV_ARC.1 refers to the following text:

Developer action elements:

ADV_ARC.1.1D The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2D The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3D The developer shall provide a security architecture description of the TSF.

Content and presentation elements:

ADV_ARC.1.1C The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2C The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3C The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4C The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5C The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

Evaluator action elements:

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

Refinement

200 **The security architecture description of the TSF initialisation process shall include the procedures to establish full functionality after power-up, state transitions from the secure state as required by FPT_FLS.1 and any state transitions of power save modes if provided by the TOE.**

201 **The security architecture shall describe how the security architecture design and implementation prevent bypassing the SFRs limiting the availability of the Test Features as required by the Limited capability and availability policy defined in FMT_LIM.2. This includes any configuration of the availability of the Test Features performed by the TOE Manufacturer before TOE Delivery.**

6.2.2.6 Refinement of Functional Specification (ADV_FSP)

Introduction

202 The CC assurance component of the family ADV_FSP (Functional specification) refers to the user-visible interface and behaviour of the TSF. It is an instantiation of the TOE security functional requirements. The functional specification shall show that all the TOE security functional requirements are addressed. It is the basis of the test coverage analysis (ATE_COV).

203 In the case of a Security IC, specific design mechanisms, which are non-functional in nature, provide security and additionally, a test tool is delivered to the user as a part of the TOE. Therefore, refinements are provided.

204 The intended user of the TOE is the Developer of the Security IC Embedded Software and the Composite TOE Manufacturer (cf. paragraph 175).

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

Developer action elements:

ADV_FSP.4.1D The developer shall provide a functional specification.

ADV_FSP.4.2D The developer shall provide a tracing from the functional specification to the SFRs.

Content and presentation elements:

ADV_FSP.4.1C The functional specification shall completely represent the TSF.

ADV_FSP.4.2C The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.4.3C The functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.4.4C The functional specification shall describe all actions associated with each TSFI.

ADV_FSP.4.5C The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

ADV_FSP.4.6C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

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

ADV_FSP.4.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

Refinement

206 Although the IC Dedicated Test Software is part of the TOE, the Test Features of the IC Dedicated Test Software are not described in the functional specification because the IC Dedicated Test Software is considered as a test tool delivered with the TOE but not providing security functionality for the operational phase of the TOE.

207 **The functional specification shall trace also security features that do not provide any external interface but that contribute to fulfil the SFRs e.g. like physical protection. Thereby they are part of the complete instantiation of the SFRs.**

208 **The functional specification is expected to refer to mechanisms against physical attacks in a general way but detailed enough to support the test coverage analysis of those mechanisms where the inspection of the layout is relevant or tests besides the TSFI are needed.**

209 **The functional specification shall specify the operating conditions of the TOE. These conditions include but are not limited to the frequency of the clock, the power supply, and the temperature.**

Background information

210 The functional specification covers all functions and mechanisms which control access to the functions provided by the IC Dedicated Test Software (refer to the security functional requirement FMT_LIM.2). Details are described according to all relevant requirements of the CC assurance class ADV including ADV_ARC.1 document, refer to 6.2.2.5, because these functions and mechanisms are active after TOE Delivery and are subject to functional testing (class ATE) and vulnerability analysis (class AVA). These descriptions are necessary inputs for testing and vulnerability assessment.

6.2.2.7 Refinement of Implementation Representation (ADV_IMP)

Introduction

211 The CC assurance component of the family ADV_IMP (implementation representation) refers to the implementation representation of the TSF. Since most parts of the

Security IC are security enforcing it is expected that the complete implementation representation is available for the evaluators[16].

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

Developer action elements:

ADV_IMP.1.1D The developer shall make available the implementation representation for the entire TSF.

ADV_IMP.1.2D The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.

Content and presentation elements:

ADV_IMP.1.1C The implementation representation shall define the TSF to a level of detail such that the TSF may be generated without further design decisions.

ADV_IMP.1.2C The implementation representation shall be in the form used by the development personnel.

ADV_IMP.1.3C The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

Evaluator action elements:

ADV_IMP.1.1E The evaluator shall confirm that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

Refinement

213 **It shall be checked that the provided implementation representation is complete and sufficient to ensure that analysis activities are not curtailed due to lack of information.**

6.2.2.8 Refinement of Test Coverage (ATE_COV)

Introduction

214 The CC 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.

[16] As stated in [3]: "The entire implementation representation is made available to ensure that analysis activities are not curtailed due to lack of information. This does not, however, imply that all of the representation is examined when the analysis activities are being performed."

215 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 of evidence 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

216 **The TOE shall 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)” shall be enforced for the complete TSF. The tests shall cover functions which may be affected by “ageing” (such as NVM writing).**

217 **The existence and effectiveness of mechanisms against physical attacks (as specified by the functional requirement FPT_PHP.3) cannot be functionally tested in a straightforward way. Therefore, the TOE Manufacturer shall provide evidence, e.g. layout design principles and actual implementation, that the TOE has the expected physical characteristics. The layout shall be checked in an appropriate way. The required evidence pertains to the existence of mechanisms against physical attacks (unless these are obvious).**

Background information

218 The IC Dedicated Test Software is seen as a test tool that is delivered as part of the TOE. However, the Test Features do not provide security functionality and are therefore not in the scope of the test coverage analysis. Nevertheless, all functions and mechanisms which limit the capability of the functions (cf. FMT_LIM.1) and control access to the functions (cf. FMT_LIM.2) provided by the IC Dedicated Test Software are in the scope of the test coverage analysis.

6.2.2.9 Refinement of User Guidance (AGD_OPE)

Introduction

219 The CC assurance components of the families AGD_OPE (Operational user guidance), and AGD_PRE (Preparative procedures) describe all relevant aspects for the secure application of the TOE.

220 The Operational User Guidance documents should provide only the information which is necessary for using the TOE. Depending on the recipients, the Operational User Guidance and Preparative Procedures can be given in the same document.

221 After production, the TOE is tested through the contact pads or any other physical interface that usually become part of the interface during packaging. No guidance document according to CC class AGD is required provided that the tests are performed by the TOE Manufacturer. Note that test procedures are described under the CC assurance component of the family ATE_FUN.

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

Developer action elements:

AGD_OPE.1.1D The developer shall provide 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 controls 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

223 **The TOE serves as a platform for the Security IC Embedded Software. Therefore, the role of the Developer of the Security IC Embedded Software is the focus of the guidance, refer also to paragraph 175.**

224 **If the TOE provides security functionality which (i) can or need to be administrated by the Security IC Embedded Software or (ii) if the IC Dedicated Support Software provides additional services (refer to 1.3), these aspects shall be described in the guidance. This may also comprise specific functionality that must be provided by the Security IC Embedded Software to support the security of the platform and configuration options of the TOE.**

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

Background information

226 Most of the security functionality may be effective before TOE Delivery. However, guidance to determine, disable, enable or modify the behaviour of security functionality, is necessary if it is possible to configure the TOE after delivery, either by the Developer of the Security IC Embedded Software or by the Composite Product Manufacturer. This guidance is delivered by the TOE Manufacturer.

6.2.2.10 Refinement of Preparative Procedures (AGD_PRE)

Introduction

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

228 The following text reflects specific 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 the 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

229 **For the Security IC, the delivery acceptance procedures comprise procedures to identify the TOE and eventually verify the authenticity of the hardware using e.g. the security functionality provided according to FAU_SAS.1.**

230 **The TOE may be configured after production before the Composite Product is delivered to the End-consumer. In this case, these configuration aspects shall be considered. Differences between the TOE before its first use, e.g. during wafer test, and Phase 7 shall be summarised. Procedures to change that behaviour shall exist.**

231 **The preparation may include the download of Security IC Embedded Software if parts of the Security IC Embedded Software are stored in the programmable NVM. If the TOE includes IC Dedicated Support Software that is delivered separately, the preparative procedures shall cover the integration of this software. The preparative procedures shall also cover the configuration of the TOE according to the options described in the ST that can be changed after TOE Delivery.**

6.2.2.11 Refinement of Vulnerability Analysis (AVA_VAN).

Introduction

232 The CC assurance component AVA_VAN.5 (Advanced methodical vulnerability analysis) refers to a “methodical vulnerability analysis” which “is performed by the evaluator to ascertain the presence of potential vulnerabilities.”

233 Since [5] only provides a generic methodical approach for vulnerability analysis, specific supporting documents for IC, smartcards and similar devices shall be used, including [10].

234 The following text reflects the requirements of the selected component AVA_VAN.5:

Developer action elements:

AVA_VAN.5.1D The developer shall provide the TOE for testing.

AVA_VAN.5.2D The developer shall provide a list of third party components included in the TOE and the TOE delivery.

Content and presentation elements:

AVA_VAN.5.1C The TOE shall be suitable for testing.

AVA_VAN.5.2C The list of third party components shall include components provided by third parties, and that are part of the TOE or otherwise part of the TOE delivery.

Evaluator action elements:

AVA_VAN.5.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.5.2E The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE the components in the list of third party components, and specific IT products in the environment that the TOE depends on.

AVA_VAN.5.3E The evaluator shall perform an independent, methodical vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.

AVA_VAN.5.4E The evaluator shall conduct penetration testing based on the identified potential vulnerabilities to determine that the TOE is resistant to attacks performed by an attacker possessing High attack potential.

Refinement

235 **The vulnerability analysis shall include the justification for the rating of the TOE information that is available to an attacker and the usage of open samples since their protection is required per the refinement of ALC_DVS, cf. 6.2.2.2.**

Application Note 24: The evaluator may assess the ROM content protection in addition to the vulnerability analysis related to the SFR FDP_SDC.1 in order to assess the effectiveness of the security architecture if relevant security features of the TOE are identified and to support composite evaluation of the smartcard.

Application Note 25: The mandatory document [10] is expected to be updated regularly to match the evolution of the attack methods on smartcards. Therefore, the ST author should indicate the version of this document used for the vulnerability analysis.

Application Note 26: The vulnerability analysis covers the resistance against side channel attacks to meet the SFP “Data Processing Policy” defined for the SFR “Subset information flow control (FDP_IFC.1)” and the security architecture aspect non-bypassability of the SFR “Stored data confidentiality (FDP_SDC.1)”.

Application Note 27: The vulnerability analysis covers the potential abuse of the functions provided by the IC Dedicated Test Software after TOE Delivery (refer to FMT_LIM.1 and FMT_LIM.2 in 6.1.3). More precisely, the vulnerability analysis addresses the capability and availability of Test Features and the way they are limited so that they do not allow software to be reconstructed and/or substantial information about construction of TSF to be gathered which may enable other attacks.

6.3 Security Requirements Rationale

6.3.1 Rationale for the Security Functional Requirements

236 Table 6-1 provides an overview of how the security functional requirements are combined to meet the security objectives.

Table 6-1: Security Requirements versus Security Objectives

Objective	TOE Security Functional and Assurance Requirements
O.Leak-Inherent	- FDP_ITT.1 “Basic internal transfer protection” - FPT_ITT.1 “Basic internal TSF data transfer protection” - FDP_IFC.1 “Subset information flow control”
O.Phys-Probing	- FDP_SDC.1 “Stored data confidentiality” - FPT_PHP.3 “Resistance to physical attack”
O.Malfunction	- FRU_FLT.2 “Limited fault tolerance - FPT_FLS.1 “Failure with preservation of secure state”
O.Phys-Manipulation	- FDP_SDI.2 “Stored data integrity monitoring and action” - FPT_PHP.3 “Resistance to physical attack”
O.Leak-Forced	All requirements listed for O.Leak-Inherent - FDP_ITT.1, FPT_ITT.1, FDP_IFC.1 plus those listed for O.Malfunction and O.Phys-Manipulation - FRU_FLT.2, FPT_FLS.1, FPT_PHP.3
O.Abuse-Func	- FMT_LIM.1 “Limited capabilities” - FMT_LIM.2 “Limited availability” plus those listed for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced - FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1
O.Identification	- FAU_SAS.1 “Audit storage”

Objective	TOE Security Functional and Assurance Requirements
O.RND	- FCS_RNG.1 “Random number generation” plus those listed for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced - FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1

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

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

239 It is possible that the TOE needs additional support by the Security IC Embedded Software (e.g. timing attacks are possible if the processing time of algorithms implemented in the software depends on the content of secrets). This support shall be addressed in the guidance (AGD_OPE). Together with this, FPT_ITT.1, FDP_ITT.1 and FDP_IFC.1 are suitable to meet the objective.

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

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

242 It is possible that the TOE needs additional support by the Security IC Embedded Software (e.g. to send data over certain buses only with appropriate precautions). This support shall be addressed in the guidance (AGD_OPE). Together with this, FPT_PHP.3 is suitable to meet the objective.

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

244 The definition of this objective shows that it covers situations where the malfunction of the TOE is caused by certain operating conditions (while direct manipulation of the TOE is covered O.Phys-Manipulation). There are two possibilities: either the operating conditions are inside the tolerated range or at least one of them is outside of this range. The second case is covered by FPT_FLS.1, which states that a secure state is preserved in this case. The first case is covered by FRU_FLT.2, which states that the TOE operates correctly under normal (tolerated) conditions. The implementation behind FRU_FLT.2 and FPT_FLS.1 shall work independently so that their operation

cannot be affected by the Security IC Embedded Software (refer to the refinement). Therefore, there is no specific conditions under O.Malfunction that are not covered.

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

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

247 It is possible that the TOE needs additional support by the Security IC Embedded Software (for instance by implementing FDP_SDI.1 to check data integrity with the help of appropriate checksums, refer to 6.1.4). This support shall be addressed in the guidance (AGD_OPE). Together with this, FPT_PHP.3 is suitable to meet the objective.

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

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

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

251 This objective states that abuse of functions (especially provided by the IC Dedicated Test Software, for instance in order to read secret data) shall not be possible in Phase 7 of the life cycle. There are two possibilities to achieve this: (i) either using the functions would not be of relevance for an attacker (i.e. their capabilities are limited) or (ii) the functions cannot be used by an attacker (i.e. their availability is limited). The first possibility is specified by FMT_LIM.1 and the second one by FMT_LIM.2. Since these requirements are combined to enforce the policy, which is suitable to fulfil O.Abuse-Func, both security functional requirements together are suitable to meet the objective.

252 Other security functional requirements which prevent attackers from circumventing the functions implementing these two security functional requirements (for instance by manipulating the hardware) also support the objective. The relevant objectives are also listed in Table 6-1.

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

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

255 The definition of an extended family and component FAU_SAS.1 (instead of using a security functional requirement from Part 2 of the CC) is due to the following reason: the security functional requirement FAU_GEN.1 requires the TOE to generate the audit data and gives details on the content of the audit records (for instance, data and time). The possibility to use the functions to store security relevant data which are generated outside of the TOE is not covered by the family FAU_GEN or by other families in Part 2. Moreover, the TOE cannot add time information to the records, because it has no real-time clock.

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

257 FCS_RNG.1 requires the TOE to provide random numbers of good quality. The specification of the exact metric is left to the ST author for a specific TOE.

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

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

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

6.3.2 Dependencies of the Security Functional Requirements

261 Table 6-2 lists the security functional requirements defined in this PP, their dependencies and whether they are satisfied in this PP. The text following the table discusses the unsatisfied dependencies.

Table 6-2: Dependencies of the Security Functional Requirements

Security Functional Requirement	Dependencies	Fulfilled by security requirements in this PP
FRU_FLT.2	FPT_FLS.1	Yes
FPT_FLS.1	None	No dependency

Security Functional Requirement	Dependencies	Fulfilled by security requirements in this PP
FMT_LIM.1	FMT_LIM.2	Yes
FMT_LIM.2	FMT_LIM.1	Yes
FAU_SAS.1	None	No dependency
FPT_PHP.3	None	No dependency
FDP_ITT.1	FDP_ACC.1 or FDP_IFC.1	Yes
FDP_IFC.1	FDP_IFF.1	See rationale below
FPT_ITT.1	None	No dependency
FDP_SDC.1	None	No dependency
FDP_SDI.2	None	No dependency
FCS_RNG.1	None	No dependency

262 Part 2 of the CC defines the dependency of FDP_IFC.1 (Subset information flow control) on FDP_IFF.1 (Simple security attributes). This PP does not include FDP_IFF.1 as it 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 no attributes are necessary. The security functional requirement for the TOE is sufficiently described using FDP_ITT.1 and its Data Processing Policy (FDP_IFC.1).

263 As shown in Table 6-2, all other dependencies of functional requirements are fulfilled by security requirements defined in this PP.

264 The discussion in 6.3.1 has shown how the security functional requirements support each other in meeting the security objectives of this PP. In particular, the security functional requirements providing resistance of the hardware against manipulations (e.g. FPT_PHP.3) support all other more specific security functional requirements (e.g. FCS_RNG.1) because they prevent an attacker from disabling or circumventing the latter.

6.3.3 Rationale for the Security Assurance Requirements

265 The assurance level EAL4 augmented with the requirements ALC_DVS.2, ALC_FLR.2 and AVA_VAN.5 meets assurance expectations for the TOE type defined in this PP as explained in the following paragraphs.

266 EAL4 augmented with ALC_DVS.2, ALC_FLR.2 and AVA_VAN.5 is required for this type of TOE which is intended to defend against sophisticated attacks. This assurance package permits a developer to gain maximum assurance from positive security engineering based on good commercial practices. Moreover, access to the low-level design and source code by evaluators ensures a suitable level of assurance that the TOE provides an adequate level of defence against such attacks.

ALC_DVS.2 Sufficiency of security controls

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

268 In the case of a Security IC, the TOE is developed and produced within a complex and distributed industrial process which shall be protected. Details about the implementation, (e.g. from design, test and development tools as well as Initialisation Data) may make attacks easier. Therefore, for a Security IC, maintaining the confidentiality of the design is very important.

269 This assurance component is a higher hierarchical component to EAL4, which includes ALC_DVS.1. ALC_DVS.2 has no dependencies.

ALC_FLR.2 Flaw reporting procedures

270 The TOE is expected to host sensitive Embedded Software which requires tracking and remediation of any reported security flaws. The TOE Developer is expected to define and use procedures to ensure the timely reception and management of flaw reports and the communication of associated corrective measures and fixes.

271 ALC_FLR.2 provides sufficient assurance about flaw remediation and reporting procedures applicable to any release of the TOE. ALC_FLR.2 does not have any dependency.

AVA_VAN.5 Advanced methodical vulnerability analysis

272 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.

273 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.

274 AVA_VAN.5 has dependencies on ADV_ARC.1 “Security architecture description”, ADV_FSP.4 “Complete functional specification”, ADV_TDS.3 “Basic modular design”, ADV_IMP.1 “Implementation representation of the TSF”, AGD_OPE.1 “Operational user guidance”, AGD_PRE.1 “Preparative procedures” and ATE_DPT.1 “Testing: basic design”.

275 All these dependencies are satisfied in EAL4.

276 It is assumed that attackers with high attack potential try to attack Security ICs like smartcards used for digital signature applications or payment systems. Therefore, specifically AVA_VAN.5 was chosen to ensure that such attackers cannot successfully attack the TOE.

277 Note that details of the refinement of the assurance requirements are given in 6.2.2.

6.3.4 Security Requirements Internal Consistency Rationale

278 The discussion of security functional requirements and assurance components in the preceding sections has shown the consistency of both groups of requirements. The arguments given for the fact that the assurance components are adequate for the functionality of the TOE also shows that the security functional requirements and assurance requirements support each other and that there are no inconsistencies between these groups.

279 The security functional requirements FDP_SDC.1 and FDP_SDI.2 address the protection of user data in the specified memory areas against compromise and manipulation. The security functional requirement FPT_PHP.3 makes it harder to manipulate data. This protects the primary assets identified in 3.1 and other security features or functionality which use these data.

280 Though a manipulation of the TOE (refer to FPT_PHP.3) is not of great value for an attacker in itself, it can be an important step in order to threaten the primary assets identified in 3.1. Therefore, the security functional requirement FPT_PHP.3 is not only required to meet the security objective O.Phys-Manipulation. Instead, it protects other security features or functions of both the TOE and the Security IC Embedded Software from being bypassed, deactivated or changed. In particular, this may pertain to the security features or functions being specified using FDP_ITT.1, FPT_ITT.1, FPT_FLS.1, FMT_LIM.2, FCS_RNG.1, and those implemented in the Security IC Embedded Software.

281 A malfunction of TSF (refer to FRU_FLT.2 and FPT_FLS.1) can be an important step in order to threaten the primary assets identified in 3.1. Therefore, the security functional requirements FRU_FLT.2 and FPT_FLS.1 are not only required to meet the security objective O.Malfunction. Instead, they protect other security features or functions of both the TOE and the Security IC Embedded Software from being bypassed, deactivated or changed. This pertains to the security features or functions being specified using FDP_ITT.1, FPT_ITT.1, FMT_LIM.1, FMT_LIM.2, FCS_RNG.1, and those implemented in the Security IC Embedded Software.

282 In a forced leakage attack the methods described in “Malfunction due to Environmental Stress” (refer to T.Malfunction) and/or “Physical Manipulation” (refer to T.Phys-Manipulation) are used to cause leakage from signals which normally do not contain significant information about secrets. Therefore, in order to avert the disclosure of primary assets identified in 3.1 it is important that the security functional requirements averting leakage (FDP_ITT.1 and FPT_ITT.1) and those against malfunction (FRU_FLT.2 and FPT_FLS.1) and physical manipulation (FPT_PHP.3) are effective and bind well. The security features and functions against malfunction ensure the correct operation of other security functions (cf. previous discussion) and help to avert forced leakage in other attack scenarios. The security features and functions against physical manipulation make it harder to manipulate the other security functions (cf. previous discussion).

283 Physical probing (refer to FPT_PHP.3) shall directly avert the disclosure of primary assets identified in 3.1. In addition, physical probing can be an important step in other attack scenarios if the corresponding security features or functions use secret data. For instance, the security functional requirement FMT_LIM.2 may use passwords. Therefore, the security functional requirement FPT_PHP.3 (against probing) helps to protect other security features or functions including those being implemented in the Security IC Embedded Software. Details depend on the implementation.

284 Leakage (refer to FDP_ITT.1 and FPT_ITT.1) directly concerns the disclosure of primary assets identified in 3.1. In addition, inherent leakage and forced leakage (cf. previous discussion) can be an important step in other attack scenarios if the corresponding security features or functions use secret data. For instance, the security functional requirement FMT_LIM.2 may use passwords. Therefore, the security functional requirements FDP_ITT.1 and FPT_ITT.1 help to protect other security features or functions implemented in the Security IC Embedded Software (FDP_ITT.1) or provided by the TOE (FPT_ITT.1). Details depend on the implementation.

285 The user data of the Composite TOE are treated as required to meet the requirements defined for the specific application context (refer to A.Resp-Appl 'Treatment of user data of the Composite TOE'). However, the TOE may implement additional functions. This can be a risk if their interface cannot completely be controlled by the Security IC Embedded Software. Therefore, the security functional requirements FMT_LIM.1 and FMT_LIM.2 are very important. They ensure that appropriate control is applied to the interface of these functions (limited availability) and that these functions, if they are usable, provide limited capabilities only.

286 The combination of the security functional requirements FMT_LIM.1 and FMT_LIM.2 ensures that (especially after TOE Delivery) these additional functions cannot be abused by an attacker to (i) disclose or manipulate user data of the Composite TOE, (ii) to manipulate (explore, bypass, deactivate or change) security features or services of the TOE or of the Security IC Embedded Software or (iii) to enable other attacks on the assets. Hereby, the binding between these two security functional requirements is very important.

287 The security functional requirement Limited Capabilities (FMT_LIM.1) closes gaps which could be left by the control being applied to the function's interface (Limited Availability (FMT_LIM.2)). Note that the security features or services which limit the availability can be bypassed, deactivated or changed by physical manipulation or a malfunction caused by an attacker. Therefore, if Limited Availability (FMT_LIM.2) is vulnerable[17], it is important to limit the capabilities of the functions in order to limit any possible benefit for an attacker.

288 The security functional requirement Limited Availability (FMT_LIM.2) closes gaps which could result from the fact that the function's kernel in principle would allow to perform attacks. The TOE shall limit the availability of functions which potentially provide the capability to disclose or manipulate user data of the Composite TOE, to manipulate security features or services of the TOE or of the Security IC Embedded Software or to enable other attacks on the assets. Therefore, if an attacker could benefit from using such functions[18], it is important to limit their availability so that an attacker is not able to use them.

289 No perfect solution to limit the capabilities (FMT_LIM.1) is required if the limited availability (FMT_LIM.2) alone can prevent the abuse of functions. No perfect solution to limit the availability (FMT_LIM.2) is required if the limited capabilities (FMT_LIM.1)

[17] Or, in the extreme case, not being provided.

[18] The capabilities are not limited in a perfect way (FMT_LIM.1).

alone can prevent the abuse of functions. Therefore, it is correct that both requirements are defined in a way that they together provide sufficient security.

290 It is important to avert malfunctions of TSF and of security functions implemented in the Security IC Embedded Software (cf. previous discussion). There are two security functional requirements which ensure that malfunctions cannot be caused by exposing the TOE to environmental stress. First, it shall be ensured that the TOE operates correctly within some limits (Limited fault tolerance (FRU_FLT.2)). Second, the TOE shall prevent its operation outside these limits (Failure with preservation of secure state (FPT_FLS.1)). These security functional requirements together prevent malfunctions. The two functional requirements shall define the "limits". Otherwise, there could be some ranges of operating conditions which are not covered so that malfunctions may occur. Consequently, the security functional requirements Limited fault tolerance (FRU_FLT.2) and Failure with preservation of secure state (FPT_FLS.1) are defined in a way that they together provide sufficient security.

7 Functional Packages

7.1 Package "Authentication of the Security IC"

7.1.1 Identification

Title:	Authentication of the Security IC
Version:	0.3 (draft)
Date:	13/08/2025
CC Edition:	CC:2022 Revision 1
Package type:	Functional

7.1.2 Overview

291 This package enhances the unique identification of the TOE, with respect to authentication by external entities.

292 This package is optional. It defines specific SPD, security objectives and an elective SFR. An ST should include this package if the reliable identification of the TOE is required and the masquerade of the genuine TOE is a threat in the End-usage operational environment (Phase 7).

7.1.3 Security Problem Definition, Security Objectives and Rationale

293 This package defines the threat "Masquerade the TOE (T.Masquerade_TOE)" as specified below.

T.Masquerade_TOE **Masquerade the TOE**

An attacker may threaten the property being a genuine TOE by producing a chip which is not a genuine TOE but wrongly identifying itself as genuine TOE sample.

The threat T.Masquerade_TOE may threaten the unique identity of the TOE or the property as being a genuine TOE without a unique identity. Mitigation of masquerade requires tightening up the identification by authentication.

294 The TOE shall provide “Authentication to external entities (O.Authentication)” as specified below.

O.Authentication **Authentication to external entities**

The TOE shall be able to authenticate itself to external entities. The Initialisation Data (or parts of them) are used as TOE authentication reference data.

295 The operational environment shall provide “External entities authenticating of the TOE (OE.TOE_Auth)”.

OE.TOE_Auth **External entities authenticating of the TOE**

The operational environment shall support the authentication verification mechanism and know the authentication reference data of the TOE.

296 The threat “Masquerade the TOE (T.Masquerade_TOE)” is directly covered by the TOE security objective “Authentication to external entities (O.Authentication)” describing the proving part of the authentication and the security objective for the operational environment of the TOE “External entities authenticating of the TOE (OE.TOE_Auth)” the verifying part of the authentication.

7.1.4 Security Functional Requirements and Rationales

297 The TOE shall meet the requirement “Authentication Proof of Identity (FIA_API.1)” as specified below.

FIA_API.1 **Authentication Proof of Identity**

Hierarchical to: No other components.

Dependencies: No dependencies.

FIA_API.1.1 The TSF shall provide an [assignment: *authentication mechanism*] to prove the identity of the TOE[19] by including the following properties [assignment: *list of properties*] to an

[19] [assignment: entity]

external entity.

Application Note 28: The ST author shall perform the open assignment in the element FIA_API.1.1. The authentication mechanism to be assigned could be a cryptographic mechanism based on a key stored in protected memory of the TOE. The selection “TOE” defines the identity as authentic example of TOE (cf. O.Authentication) which is authenticated to an external entity (e.g. the Composite Product Manufacturer or the Personalisation agent). The proved identity depends on the set of TOE samples holding the same authentication verification data and the identity linked to the authentication reference data. E.g. the update of Security IC Embedded Software in life cycle Phase 7 may need chip-individual private keys and chip-individual certificates of the corresponding public keys. In other use cases (e.g. electronic passports) the protection of personal data may be a concern and requiring many chips having the same private key.

298 The security objective “Authentication to external entities (O.Authentication)” is directly covered by FIA_API.1.

299 FIA_API.1 has no dependencies.

7.2 Packages for Loaders

300 This section describes the security requirements for a Loader provided by the TOE. The Loader may be used to load data into the NVM after delivery of the TOE. This Loader is considered as part of the TOE and is associated with the IC Dedicated Support Software (cf. para. 17).

301 The IC Manufacturer may install Configuration Data, Initialisation Data and IC Dedicated Software and may be required by the Composite Product Integrator to install Security IC Embedded Software or other user data during the manufacturing process. The manufacturing tools and processes used are not available after TOE Delivery and therefore they are not associated with the Loader packages but covered by the ALC class.

302 The Security IC Embedded Software may implement its own mechanism for loading data into NVM, but this functionality is out of scope of the Loader packages.

303 The loaded data may be of different type and owner:

- IC Dedicated Support Software as part of the current TOE or a new TOE, or
- user data of the TOE as the Security IC Embedded Software, TSF data or user data of the Composite Product, e.g. a smartcard product.

304 The Loader may be used in different operational environments of the TOE:

- Secured environments maintain the confidentiality and integrity of the TOE as required by OE.Process-Sec-IC and the confidentiality and integrity of the Security IC Embedded Software, TSF data or user data associated with the Composite Product by security procedures of the Composite Product Manufacturer, personaliser and other actors before delivery to the end-user depending on the product life cycle.

- Operational environments including “Phase 7 Security IC End-usage” require self-protected TOEs which control the access to the Loader and protects the loaded data. The authorized user like IC Manufacturer, the Composite Product Manufacturer, personaliser or Issuer needs reliable identification of the TOE for loading or modification of IC Dedicated Support Software, Security IC Embedded Software, TSF data or user data of the Composite Product.

305 The packages address different functionality and method of use of the Loader:

- Package Loader 1 “Loader dedicated for usage in Secured Environment only”
    - limited capability of the Loader protecting user data in the writable memory areas,
    - blocking of the Loader after intended usage (e.g. delivery to end-customer before Security IC End-usage phase) addressed by limited availability.
- Package Loader 2 “Loader dedicated for usage by authorized users only”
    - protection of the TOE user data against misuse of the Loader,
    - trusted channel between the Security IC and the authorised role to change the user data by means of the Loader,
    - checking the integrity and the authenticity of the data provided by the authorized user to the Loader,
    - access control to the Loader functionality.

306 The Package Loader 1 comprises a baseline set of security functionality of the Loader and assumes the usage of the Loader in secured environments. It requires the Composite Product Manufacturer to enable the protection against misuse of the Loader after its intended usage and before delivery to the end-user in life cycle Phase 7.

307 The Package Loader 2 is aimed at use cases where the users have different security policies and authorization to load or modify data in the writable memory. E.g. the intended usage of the TOE may limit the loading of IC Dedicated Support Software to users authorized by IC Manufacturer and the loading of Security IC Embedded Software to the Composite Product Integrator. The Loader may not distinguish between data as being IC Dedicated Support Software or Security IC Embedded Software but - as minimum – data as targeted to specific to memory areas. The Package Loader 2 may be useful for secured environments as well.

7.2.1 Package Loader 1 “Loader dedicated for usage in secured environment only”

7.2.1.1 Identification

Title: Loader dedicated for usage in secured environment only

Version:	0.3 (draft)
Date:	13/08/2025
CC Edition	CC:2022 Revision 1
Package type:	Functional

7.2.1.2 Overview

308 This package is intended for loaders that are used exclusively in secured environments which are controlled by the owner of the loaded data or its subcontractor. This is typically the Composite Product Manufacturer or the IC Packaging Manufacturer (cf. 1.3.5 and 11.1.2 for details).

309 This package is optional. It defines specific SPD, security objectives and a set of elective SFRs. An ST shall include all or none of the SFRs and associated SPD and objectives.

7.2.1.3 Security Problem Definition, Security Objectives and Rationale

310 The threat “Abuse the Loader Functionality (T.Abuse_Loader1)” applies to loaders dedicated for usage in secured environment.

T.Abuse_Loader1 **Abuse the Loader Functionality**

An attacker may use the Loader functionality after TOE Delivery to manipulate (explore, bypass, deactivate or change) the security services of the TOE that are provided by or depend on the IC Dedicated Support Software, TSF data, the Security IC Embedded Software, or user data of the Composite Product.

311 The organisational security policy “Limiting and Blocking the Loader Functionality (P.Lim_Block_Loader1)” applies to loaders dedicated for usage in secured environment.

P.Lim_Block_Loader1 **Limiting and Blocking Loader Functionality**

The Composite Product Manufacturer uses the Loader for loading the Security IC Embedded Software, user data of the Composite Product or IC Dedicated Support Software in charge of the IC Manufacturer. He limits the capability and blocks the availability of the Loader in order to protect stored data from disclosure and manipulation.

312 The TOE shall provide “Capability and availability of the Loader (O.Cap_Avail_Loader1)” as specified below.

O.Cap_Avail_Loader1 **Capability and availability of the Loader**

The TOE shall provide limited Loader capabilities and irreversible termination of the Loader in order to protect stored user data from disclosure and manipulation.

313 The operational environment of the TOE shall provide “Limitation of capability and blocking the Loader (OE.Lim_Block_Loader1)” as specified below.

OE.Lim_Block_Loader1 **Limitation of capability and blocking the Loader**

The Composite Product Manufacturer shall protect the Loader functionality against misuse, limit the capability of the Loader and terminate irreversibly the Loader after intended usage of the Loader.

314 The organisational security policy Limitation and Blocking the Loader Functionality (P.Lim_Block_Loader1) is directly implemented by the security objective for the TOE “Capability and availability of the Loader (O.Cap_Avail_Loader1)” and the security objective for the TOE environment “Limitation of capability and blocking the Loader (OE.Lim_Block_Loader1)”.

315 The TOE security objective “Capability and availability of the Loader” (O.Cap_Avail_Loader1)” mitigates also the threat “Abuse of Loader Functionality “ (T.Abuse_Loader1) if attacker tries to misuse the Loader functionality in order to manipulate security services of the TOE provided or depending on IC Dedicated Support Software or user data of the TOE as Security IC Embedded Software, TSF data or user data of the Composite Product.

7.2.1.4 Security Functional Requirements and Rationales

316 The TOE shall meet the requirement “Limited capabilities – Loader (FMT_LIM.1/Loader1)” as specified below.

FMT_LIM.1/Loader1 **Limited capabilities**

Hierarchical to: No other components.

Dependencies: FMT_LIM.2 Limited availability.

FMT_LIM.1.1/Loader1 The TSF shall limit its capabilities so that in conjunction with “Limited availability (FMT_LIM.2)” the following policy is enforced: Deploying Loader functionality after [assignment: *action*] does not allow stored user data to be disclosed or manipulated by unauthorized users[23].

Application Note 29: FMT_LIM.1 supplements FMT_LIM.2 allowing for non-overlapping loading of user data and protecting the TSF against misuse of the Loader for attacks against the TSF. The TOE Loader may allow for correction of already loaded user data before the assigned action e.g. before blocking the TOE Loader for TOE Delivery to the end-customer or any intermediate step in the life cycle of the Security IC or the smartcard.

317 The TOE shall meet the requirement “Limited availability – Loader (FMT_LIM.2/Loader1)” as specified below.

FMT_LIM.2/Loader1 Limited availability

Hierarchical to: No other components.

Dependencies: FMT_LIM.1 Limited capabilities.

FMT_LIM.2.1/Loader1 The TSF shall be designed in a manner that limits its availability so that in conjunction with “Limited capabilities (FMT_LIM.1)” the following policy is enforced: The TSF prevents deploying the Loader functionality after [assignment: *action*][20].

Application Note 30: This is the easiest variant of Loader functionality relying on secure boot loading procedures in a secured environment before TOE Delivery to the assigned customer and preventing deploying the Loader of the Security IC after the assigned action, e.g. after blocking of Loader for TOE Delivery to the End-customer.

318 The security objective “Capability and availability of the Loader (O.Cap_Avail_Loader1) is directly covered by the SFRs FMT_LIM.1/Loader1 and FMT_LIM.2/Loader1.

319 The SFR dependencies are satisfied as shown in Table 7-1

Table 7-1: Package Loader 1 - SFR Dependencies

Security Functional Requirement	Dependencies	Fulfilled by
FMT_LIM.1/Loader1	FMT_LIM.2	FMT_LIM.2/Loader1
FMT_LIM.2/Loader1	FMT_LIM.1	FMT_LIM.1/Loader1

7.2.2 Package Loader 2 “Loader dedicated for usage by authorized users only”

7.2.2.1 Identification

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

Title:	Loader dedicated for usage by authorized users only
Version:	0.3 (draft)
Date:	13/08/2025
CC Edition:	CC:2022 Revision 1
Package type:	Functional

7.2.2.2 Overview

320 This package defines security functionality for a type of loaders that support access control, mutual authentication of the TOE and the authorized user as endpoints of a trusted channel, and protection of integrity and confidentiality of the loaded data.

321 This package is optional. It defines specific SPD, security objectives and a set of elective SFRs. An ST shall include all or none of the SFRs and associated SPD and objectives.

322 The ST shall include this package if the loader is intended to be used in Phase 7: Operational usage of the life cycle. The package may be used also if the loader is intended to be used after delivery by the TOE Manufacturer in Phase 4: IC Packaging, Phase 5: Composite Product Integration and Phase 6: Personalisation, e.g. by different authorized users.

7.2.2.3 Security Problem Definition, Security Objectives and Rationale

323 The organisational security policy "Controlled usage to Loader Functionality (P.Ctlr_Loader2)" applies to loaders dedicated for usage by authorized users only.

P.Ctlr_Loader2 **Controlled usage to Loader Functionality**

Authorized user controls the usage of the Loader functionality in order to protect stored and loaded user data from disclosure and manipulation.

324 The TOE shall provide "Access control and authenticity for the Loader (O.Ctrl_Auth_Loader2)" as specified below.

O.Ctrl_Auth_Loader2 **Access control and authenticity for the Loader**

The TOE shall provide trusted communication channel with authorized user, supports confidentiality protection and authentication of the user data to be loaded and access control for usage of the Loader functionality.

325 The operational environment of the TOE shall provide "Secure communication and usage of the Loader (OE.Usage_Loader2)" as specified below.

OE.Usage_Loader2 **Secure communication and usage of the Loader**

The authorized user shall support the trusted communication channel with the TOE by confidentiality protection and authenticity proof of the data to be loaded and fulfilling the access conditions required by the Loader.

326 The organisational security policy “Controlled usage to Loader Functionality (P.Ctlr_Loader2) is directly implemented by the security objective for the TOE “Access control and authenticity for the Loader (O.Ctrl_Auth_Loader2)” and the security objective for the TOE environment “Secure communication and usage of the Loader (OE.Usage_Loader2)”.

7.2.2.4 Security Functional Requirements and Rationales

327 The TOE shall meet the requirement “Inter-TSF trusted channel (FTP_ITC.1)” as specified below.

FTP_ITC.1/ Loader2 **Inter-TSF trusted channel**

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1/ Loader2 The TSF shall provide a communication channel between itself and **[assignment: *users authorized 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/ Loader2 The TSF shall permit another trusted IT product[21] to initiate communication via the trusted channel.

FTP_ITC.1.3/ Loader2 The TSF shall initiate communication via the trusted channel for deploying Loader [assignment: *rules*][22].

Application Note 31: FTP_ITC.1.1/Loader is a refined element which requires that the TOE only communicates with authorized the users, not with trusted IT products as the original text expresses[23]. That is, trust is not put on the IT product that is used to convey the information but, on the users, i.e. the entities behind the non-TOE end points.

[21] [selection: *the TSF, another trusted IT product*]

[22] [assignment: *list of functions for which a trusted channel is required*]

[23] The text of the element FTP_ITC.1.1 which has been refined is the following: “The TSF shall provide a communication channel between itself and another trusted IT product 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.”

328 The TOE shall meet the requirement “Basic data exchange confidentiality (FDP_UCT.1)” as specified below.

FDP_UCT.1/ Loader2 **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/ Loader2 The TSF shall enforce the Loader SFP[24] to receive[25] user data in a manner protected from unauthorised disclosure.

329 The TOE shall meet the requirement “Data exchange integrity (FDP_UIT.1)” as specified below.

FDP_UIT.1/ Loader2 **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/ Loader2 The TSF shall enforce the Loader SFP[26] to receive[27] user data in a manner protected from modification, deletion, insertion[28] errors.

FDP_UIT.1.2/ Loader2 The TSF shall be able to determine on receipt of user data, whether modification, deletion, insertion[29] has occurred.

330 The TOE shall meet the requirement “Subset access control (FDP_ACC.1)” as specified below.

FDP_ACC.1/ Loader2 **Subset access control**

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

[25] [selection: *transmit, receive*]

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

[27] [selection: *transmit, receive*]

[28] [selection: *modification, deletion, insertion, replay*]

[29] [selection: *modification, deletion, insertion, replay*]

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control.

FDP_ACC.1.1/ Loader2 The TSF shall enforce the Loader SFP[30] on
(1) the subjects [assignment: *authorized roles for using Loader*],
(2) the objects user data in [assignment: *memory areas*],
(3) the operation deployment of Loader[31]

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

FDP_ACF.1/ Loader2 **Security attribute based access control**

Hierarchical to: No other components.

Dependencies: FMT_MSA.3 Static attribute initialisation

FDP_ACF.1.1/ Loader2 The TSF shall enforce the Loader SFP[32] to objects based on the following:
(1) the subjects [assignment: a*uthorized roles for using Loader*] with security attributes [assignment: *SFP-relevant security attributes, or named groups of SFP-relevant security attributes*]
(2) the objects user data in [assignment: *memory areas*] with security attributes [assignment: *SFP-relevant security attributes, or named groups of SFP-relevant security attributes*][33].

FDP_ACF.1.2/ Loader2 The TSF shall enforce the following rules to determine if 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/ Loader2 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*].

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

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

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

[33] [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*]

FDP_ACF.1.4/ Loader2 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 32: The ST author shall perform the open operations in the component of FDP_ACF.1/Loader2 in order to describe the access control rules. In FDP_ACF.1.1/Loader2, the open assignment of security attributes may be empty.

332 The security objective "Access control and authenticity for the Loader (O.Ctrl_Auth_Loader2)" is covered by the SFRs as follows:

- The requirement FDP_ACC.1/Loader2 defines the subjects, objects and operations of the Loader SFP enforced by the SFR FTP_ITC.1/Loader2, FDP_UCT.1/Loader2, FDP_UIT.1/Loader2 and FDP_ACF.1/Loader2.

- The requirement FTP_ITC.1/Loader2 requires the TSF to establish a trusted channel with assured identification of its end points and protection of the channel data from modification or disclosure.

- The requirements FDP_UCT.1/Loader2 and FDP_UIT.1/Loader2 requires the TSF to receive data protected from unauthorised disclosure and modification

- The requirement FDP_ACF.1/Loader2 requires the TSF to implement access control for the Loader functionality.

333 The SFR dependencies are satisfied as shown in Table 7-2.

Table 7-2: Package Loader 2 - SFR Dependencies

Security Functional Requirement	Dependencies	Fulfilled by security requirements in this PP
FTP_ITC.1/Loader2	No dependency	
FDP_UCT.1/Loader2	[FTP_ITC.1, or FTP_TRP.1] [FDP_ACC.1, or FDP_IFC.1]	FTP_ITC.1/Loader2 FDP_ACC.1/Loader2
FDP_UIT.1/Loader2	[FTP_ITC.1, or FTP_TRP.1] [FDP_ACC.1, or FDP_IFC.1]	FTP_ITC.1/Loader2 FDP_ACC.1/Loader2
FDP_ACC.1/Loader2	FDP_ACF.1	FDP_ACF.1/Loader2
FDP_ACF.1/Loader2	FMT_MSA.3	See rationale below

334 The requirement FMT_MSA.3 is not included in this package because no predefined security attribute is required to enforce the Loader SFP. Any necessary security attribute shall be defined in the ST. If the set of attributes is not empty, then the ST author shall include FMT_MSA.3 to specify the management of the security attributes enforcing the Loader SFP, and any necessary further dependencies.

7.3 Package “Cryptographic Services”

7.3.1 Identification

Title:	Cryptographic Services
Version:	0.3 (draft)
Date:	13/08/2025
CC Edition:	CC:2022 Revision 1
Package type:	Functional

7.3.2 Overview

335 This package defines a generic set of requirements for the cryptographic services provided by the TOE to the Security IC Embedded Software.

336 This package is optional. It defines specific SPD, security objectives and a set of elective SFRs. An ST shall include all or none of the SFRs and associated SPD and objectives. The ST author shall iterate the SFRs as necessary to cover all the cryptographic services in the scope of the evaluation.

7.3.3 Security Problem Definition, Security Objectives and Rationale

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

P.Crypto-Service Cryptographic services of the TOE

The TOE provides secure hardware-based cryptographic services for the Security IC Embedded Software.

338 To enforce this policy, the ST author defines a specific security objective for every cryptographic service present in the TOE. The objective is fulfilled by SFRs from the Cryptographic Support (FCS) class, which shall be met by hardware-based implementation, that is purely hardware-based implementation or hybrid hardware-and-software implementation.

339 The TOE shall provide the “Cryptographic service (O.Crypto-Service)” as specified below.

O.Crypto-Service Cryptographic services

The TOE shall provide secure hardware-based cryptographic services implementing cryptographic algorithms.

Application Note 33: Encryption, decryption, signature and hashing are examples of cryptographic services. The objective O.Crypto-Service shall be integrated for each cryptographic algorithm to support the mapping to the associated Security Functional Requirements.

340 The organisational security policy ‘Cryptographic services of the TOE’ (P.Crypto-Service) is implemented directly by the TOE security objective ‘Cryptographic Algorithm’ (O.Crypto-Service).

7.3.4 Security Functional Requirements and Rationales

341 The TOE shall meet the requirement “Cryptographic operation of the selected algorithm FCS_COP.1” as specified below.

FCS_COP.1 **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.6 Timing and event of cryptographic key destruction

FCS_COP.1.1 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 34: The ST author shall iterate FCS_COP.1 as many times as necessary to specify the set of cryptographic services in the scope of the evaluation. For instance, the ST author can define one instance of FCS_COP.1 for each algorithm and specify within the same SFRs the suitable set of cryptographic operations, key sizes and standards.

Application Note 35: The cryptographic operations defined in [7] include cryptographic algorithms according to standards accepted by various certification bodies. The use of such cryptographic algorithms supports the re-use of evaluation results for higher assurance levels.

342 The TOE shall meet the requirement “Timing and event of cryptographic key destruction (FCS_CKM.6)” as specified below.

FCS_CKM.6 **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, or

FCS_CKM.5 Cryptographic key derivation]

FCS_CKM.6.1 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 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 36: The cryptographic key destruction may be provided by overwriting the internal stored key when a new key value is provided through the key interface or a key zeroize initiated by special signal.

Application Note 37: The ST author shall include as many iterations of FCS_CKM.6 as necessary, for instance by key or by key destruction method. Depending on the implemented key storage and the defined key destruction method, a single instance of FCS_CKM.6 may satisfy the dependency for multiple cryptographic algorithms defined using FCS_COP.1.

343 The requirements FCS_COP.1 and FCS_CKM.6 meet the security objective “Cryptographic service (O.Crypto-Services)”.

344 The SFR dependencies are satisfied as shown in Table 7-3.

Table 7-3: Package Cryptographic Services - SFR Dependencies

Security Functional Requirement	Dependencies	Fulfilled by security requirements in this PP
FCS_COP.1	[FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5] FCS_CKM.6	FCS_CKM.6 See rationale below
FCS_CKM.6	[FDP_ITC.1, or FDP_ITC.2, or FCS_CKM.1, or FCS_CKM.5]	See rationale below

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

346 FCS_COP.1 and FCS_CKM.6 have a dependency on [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 package does not mandate any specific source for the keys; the ST author shall include the appropriate SFR component(s) and update the dependency rationale.

7.4 Package “Address-based Access Control”

7.4.1 Identification

Title:	Address-based Access Control
Version:	0.3 (draft)
Date:	13/08/2025
CC Edition:	CC:2022 Revision 1
Package type:	Functional

7.4.2 Overview

347 This package defines a generic set of requirements for the implementation of address-based access control support.

348 The definition of restricted address areas, including memories and memory-mapped peripherals, is under the control of the Security IC Embedded Software. The Security IC enforces the partitioning of the areas so that software can only perform the permitted operations on its authorised areas, thus preserving the confidentiality and integrity of the code and data stored in these areas. Address-based access control is relevant whether the platform hosts several applications or a single application.

349 This package is optional. It defines specific SPD, security objectives and a set of elective SFRs. An ST shall include all or none of the SFRs and associated SPD and objectives.

7.4.3 Security Problem Definition, Security Objectives and Rationale

350 The TOE shall avert the threat “Address Access Violation (T.Addr-Access)” as specified below.

T.Addr-Access Address Access Violation

A part of the Security IC Embedded Software, accidentally or deliberately, accesses a memory, memory-mapped peripheral or any addressable object without authorisation, leading to a compromise (disclosure or modification) of the code or data stored at that address, or to operation disruption.

351 The TOE shall provide “Address-based Access Control (O.Addr-Access)” as specified below.

O.Addr-Access Address-based Access Control

The TOE shall allow the Security IC Embedded Software to define restricted address-based areas and shall enforce their partitioning so that only authorised operations can be performed by authorized entities within restricted areas.

Application Note 38: The address-based management services can be used by the Security IC Embedded Software for the isolation of applications in a multi-application scenario.

352 The threat 'Address Access Violation' (T.Addr-Access) is covered directly by the TOE security objective 'Address-based Access Control' (O.Addr-Access).

7.4.4 Security Functional Requirements and Rationales

353 The TOE shall meet the requirement "Subset access control (FDP_ACC.1)" as specified below.

FDP_ACC.1/Addr Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1/Addr The TSF shall enforce the Address-Based Access Control SFP[34] on

- subjects:
  - [selection: *all software, the following software parts* [assignment: *list of software parts*]] *residing in* [selection*: any memory areas, the following memory areas* [assignment: *list of memory areas*]]
  - [assignment: *list of other subjects able to request address-based operations*]
- objects:
  - code and data stored in [selection: *any memory areas, the following memory areas* [assignment: *list of memory areas*]]
  - [assignment: *list of other addressable objects*]
- operations: [assignment: *read, write, execute, and/or other operations on objects*][35].

Application Note 39: The Address-based Access Control SFP covers memory access control at least. The assignments that allow to specify other subjects and other addressable objects can be empty.

[34] [assignment: access control SFP]

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

Application Note 40: A Memory Management Unit (MMU) may or may not translate logical to physical addresses and vice versa. If it does, the term “memory area” (memory location) pertains to physical addresses because different software or data must have different attributes though perhaps being executed in the same logical address space. If it does not, i.e. no address translation is performed, memory area (memory location) pertains to both physical and logical addresses which are identical.

Application Note 41: The ST author shall specify whether memory areas pertain to (i) types of memories or (ii) address ranges or (iii) a combination of both.

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

FDP_ACF.1/Addr 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/Addr The TSF shall enforce the Address-based Access Control SFP [36] to objects based on the following permission control information:

- [selection: *the memory area where the software is executed from and/or the memory area where the access operation is performed to and/or the operation to be performed*]
- [assignment: *the SFP-relevant security attributes of any other applicable subject and object*].[37]

FDP_ACF.1.2/Addr The TSF shall enforce the following rules to determine if 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, evaluated* [assignment: *before, during and/or after each access*]].[38]

FDP_ACF.1.3/Addr The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [selection: *none,* [assignment: *list of privileged subjects and rules, based on*

[36] [assignment: access control SFP]

[37] [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]

[38] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

security attributes of the subjects, that explicitly authorize access of privileged subjects to objects]][39].

FDP_ACF.1.4/Addr The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [selection: *none*, [assignment: *list of unprivileged subjects and rules, based on security attributes of the subjects, that explicitly deny access of unprivileged subjects to objects*]] [40].

Application Note 42: In FDP_ACF.1.1, “permission control information” is a generic term denoting the set of security attributes of the SFP. The assignment can be empty if the SFP only covers memory management.

Application Note 43: In FDP_ACF.1.2, the rules shall ensure that not permitted accesses do not allow the subject to use the object, whether the evaluation of the permission is performed before, during or after the access. The following generic rule can be used to fill in the assignment provided the actual SFP rules are described in ADV documentation: “evaluate the corresponding permission control information before, during or after the access so that accesses to be denied cannot be utilised by the subject attempting to perform the operation”.

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

FMT_MSA.3/Addr 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/Addr The TSF shall enforce the Address-based Access Control SFP [41] to provide [selection: *restrictive, permissive*, [assignment: *other property*]] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2/Addr The TSF shall allow [selection: *any subject*, [assignment: *list of authorized subjects*]] (provided that the Address-Based Access Control SFP is enforced and the necessary access is therefore allowed) [42] to specify alternative initial values to override the default values when an object or information is created.

[39] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]

[40] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

[41] [assignment: access control SFP, information flow control SFP]

[42] [assignment: the authorised identified roles]

Application Note 44: The TOE is designed to support the Address-Based Access Control SFP. Therefore, the static attribute initialisation shall ensure that the SFP is enforced when control is given to the Security IC Embedded Software. The default or initial values of the permission control information (security attributes) are defined during production (FMT_MSA.3) and they are updated as required by a privileged subject (FMT_MSA.1).

Application Note 45: In FMT_MSA.3.2, the subjects play their own role (the dependency FMT_SMR.1 is not necessary).

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

FMT_MSA.1/Addr **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_MSA.1.1/Addr The TSF shall enforce the Address-based Access Control SFP[43] to restrict the ability to [selection: *change_default, modify, delete*][44] the permission control information[45] to [selection: *none,* assignment: *list of privileged subjects*]][46].

Application Note 46: Privileged subjects can be software with specific attributes.

357 The security objective “Address-based Access Control (O.Addr-Access)” is covered by the SFRs as follows:

- The requirements FDP_ACC.1/Addr and FDP_ACF.1/Addr requires that the TOE enforces the partitioning of the address space and the verification of the permission information to objects as required by O.Addr-Access.
- The requirement FMT_MSA.3/Addr requires that the TOE provides default values and overriding rules for the security attributes (permission control information). The requirement FMT_MSA.1/Addr allows to specify additional rules to update the security attributes by privileged subject(s).

358 The SFR dependencies are satisfied as shown in Table 7-4.

[43] [assignment: access control SFP(s), information flow control SFP(s)]

[44] [selection: change_default, query, modify, delete, [assignment: other operations]]

[45] [assignment: list of security attributes]

[46] [assignment: the authorised identified roles]

Table 7-4: Package Address-based Access Control - SFR Dependencies

Security Functional Requirement	Dependency	Fulfilled by
FDP_ACC.1/Addr	FDP_ACF.1	FDP_ACF.1/Addr
FDP_ACF.1/Addr	FDP_ACC.1 FMT_MSA.3	FDP_ACC.1/Addr FMT_MSA.3/Addr
FMT_MSA.3/Addr	FMT_MSA.1 FMT_SMR.1	FMT_MSA.1/Addr See rationale below
FMT_MSA.1/Addr	[FDP_ACC.1 or FDP_IFC.1] FMT_SMR.1	FDP_ACC.1/Addr See rationale below

359 The dependency FMT_SMR.1 introduced by the two components FMT_MSA.1/Addr and FMT_MSA.3/Addr is not applicable because the Address-based Access Control SFP is not role-based but enforced for subjects. Therefore, there is no need to identify roles in form of a security functional requirement FMT_SMR.1.

8 Glossary

Application Data — Data managed by the Security IC Embedded Software in the application context. Application data comprise all data in the final Security IC.

Authentication reference data — Data used to verify the claimed identity in an authentication procedure.

Authentication verification data — Data used to prove the claimed identity in an authentication procedure.

Composite Product Integrator — Role installing or finalising the Security IC Embedded Software and the applications on platform transforming the TOE into the unpersonalised Composite Product after TOE Delivery.

The TOE Manufacturer may implement IC Embedded Software delivered by the Security IC Embedded Software Developer before TOE delivery (e.g. if the IC Embedded Software is implemented in ROM or is stored in the non-volatile memory as service provided by the IC Manufacturer or IC Packaging Manufacturer).

Composite Product Manufacturer — The Composite Product Manufacturer has the following roles (i) the Security IC Embedded Software Developer (Phase 1), (ii) the Composite Product Integrator (Phase 5) and (iii) the Personaliser (Phase 6). If the TOE is delivered after Phase 3 in form of wafers or sawn wafers (dice) he has the role of the IC Packaging Manufacturer (Phase 4) in addition.

The customer of the TOE Manufacturer who receives the TOE during TOE Delivery. The Composite Product Manufacturer includes the Security IC Embedded Software developer and all roles after TOE Delivery up to Phase 6 (refer to Figure 2 and 11.1.2).

Configuration Data

TSF Data required to configure the TSF.

End-consumer

User of the Composite Product in Phase 7.

IC Dedicated Software

Software embedded in a Security IC (also known as IC firmware) and developed by the Security IC Developer. Such software is required for testing purpose (IC Dedicated Test Software) but may provide additional services to facilitate usage of the hardware and/or to provide additional services (IC Dedicated Support Software).

IC Dedicated Test Software

Part of the IC Dedicated Software which is used to test the TOE before TOE Delivery, but which does not provide any functionality thereafter.

IC Dedicated Support Software

Part of the IC Dedicated Software which provides functions after TOE Delivery. The usage of parts of the IC Dedicated Support Software may be restricted to certain phases of the product life cycle.

Initialisation Data

TSF Data defined by the TOE Manufacturer to identify the TOE and to keep track of the Security IC’s production and further life cycle phases. These data are, for instance, used for traceability and for TOE identification (Identification Data). If “Package Authentication of the Security IC” is used, the Initialisation Data contain the confidential authentication verification data of the Security IC. If the Package Loader 2 “Loader dedicated for usage by authorized users only” is used, the Initialisation Data may contain the authentication verification data or key material for the trusted channel between the TOE and the authorized users using the Loader.

Integrated Circuit (IC)

Electronic component(s) designed to perform processing and/or memory functions.

Pre-personalisation Data

Data supplied by the Composite Product Manufacturer that is injected into the NVM by the IC Manufacturer (Phase 3). These data are for instance used for traceability and/or to secure shipment between phases.

If Package Loader 2 “Loader dedicated for usage by authorized users only” is used, the Pre-personalisation Data may contain the authentication reference data or key material for the trusted channel between the TOE and the authorized users using the Loader.

Security IC

See TOE type definition in 1.3.1.

Security IC Embedded Software

Software embedded in a Security IC which is not part of the Security IC Dedicated Software. The Security IC Embedded Software is designed in Phase 1 and embedded into the Security IC in Phase 3 or in later phases of the Security IC Product life cycle.

Security IC Product

Composite Product which includes the Security IC (i.e. the TOE) and the Security IC Embedded Software and is evaluated using the composite evaluation methodology.

Secured Environment

Operational environment maintains the confidentiality and integrity of the TOE as addressed by OE.Process-Sec-IC and the confidentiality and integrity of the IC Embedded Software, TSF data or user data associated with the Composite Product by security procedures of the Composite Product Manufacturer, personaliser and other actors before delivery to the end-user depending on the product life cycle.

Test Features

All features and functions implemented by the IC Dedicated Test Software and/or hardware which are designed to be used before TOE Delivery only and delivered as part of the TOE.

TOE Delivery

The period when the TOE is delivered which is (refer to Figure 2) either (i) after Phase 3 (or before Phase 4) if the TOE is delivered in wafer or sawn wafer (dice) form factor or (ii) after Phase 4 (or before Phase 5) if the TOE is delivered in the form of a packaged product.

TOE Manufacturer

The TOE Manufacturer ensures that all requirements for the TOE (as defined in 1.3) and its development and production environment are fulfilled (refer to Figure 2).

The TOE Manufacturer has the following roles: (i) IC Developer (Phase 2) and (ii) IC Manufacturer (Phase 3). If the TOE is delivered after Phase 4 in the form of a packaged product, he has the role of the (iii) IC Packaging Manufacturer (Phase 4) in addition.

TSF data

Data for the operation of the TOE upon which the enforcement of the SFRs relies. They are created by and for the TOE and may affect the operation of the

	TOE. This includes information about the TOE's configuration, if any is coded in ROM, in NVM, in specific circuitry or a combination thereof.
User data of the Composite TOE	All data managed by the Security IC Embedded Software in the application context.
User data of the TOE	Data for the user of the TOE, that does not affect the operation of the TSF. From the point of view of TOE, the user data comprises the Security IC Embedded Software and the user data of the Composite TOE.

9 Abbreviations

CC	Common Criteria
CCRA	Common Criteria Recognition Arrangement
CEM	Common Evaluation Methodology
CM	Configuration Management
CPU	Central Processing Unit
DFA	Differential Fault Analysis
DPA	Differential Power Analysis
EAL	Evaluation Assurance Level
ETR	Evaluation Technical Report
EUCC	European Union Cybersecurity Certification
FIB	Focused Ion Beam
HDL	Hardware Description Language
HMAC	Hash-based Message Authentication Code
IC	Integrated circuit
ISO	International Organization for Standardization
IT	Information Technology
NIAP	National Information Assurance Partnership
NIST	National Institute of Standards and Technology
NVM	Non-volatile memory
PP	Protection Profile
RNG	Random number generator
ROM	Read-Only Memory
SAR	Security Assurance Requirement
SFP	Security Function Policy
SFR	Security Functional Requirement
SPA	Simple Power Analysis

SPD	Security Problem Definition
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
USB	Universal Serial Bus

10 References

[1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; November 2022, Version CC:2022, Revision 1, CCMB-2022-11-001.

[2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; November 2022, Version CC:2022, Revision 1, CCMB-2022-11-002

[3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; November 2022, Version CC:2022, Revision 1, CCMB-2022-11-003.

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

[5] Common Methodology for Information Technology Security Evaluation (CEM), Evaluation Methodology; Version CC:2022, Revision 1, CCMB-2022-11-006.

[6] Enisa, EUCC scheme, Application of Common Criteria to Integrated Circuits, Version 2 (draft), December 2024.

[7] Enisa, EUCC scheme, Guidelines on cryptography, Agreed Cryptographic Mechanisms, Version 2, May 2025.

[8] Joint Interpretation Library: Guidance for smartcard evaluation, April 2024, Version 3.0.

[9] Security architecture requirements (ADV_ARC) for smart cards and similar devices extended to Secure Sub-System in SoC, version 1.1 or latest version in force per Annex I to the COMMISSION IMPLEMENTING REGULATION 2024/482 (EUCC).

[10] Application of attack potential to smartcards and similar devices, version 1.1 or latest version in force per Annex I to the COMMISSION IMPLEMENTING REGULATION 2024/482 (EUCC).

[11] Composite product evaluation for smartcards and similar devices, version 1.1 or latest version in force per Annex I to the COMMISSION IMPLEMENTING REGULATION 2024/482 (EUCC).

[12] Joint Interpretation Library: ETR for composite evaluation TD SC & SD - Template, April 2014, Version 1.2.

[13] Minimum Site Security Requirements, version 1.1 or latest version in force per Annex I to the COMMISSION IMPLEMENTING REGULATION 2024/482 (EUCC).

[14] Eurosmart, Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, January 2014, BSI-CC-PP-0084-2014.

[15] Eurosmart, Smartcard Integrated Circuit Platform Augmentations, Version 1.00, March 2002.

[16] Evaluation of random number generators, Version 0.8, Bundesamt für Sicherheit in der Informationstechnik.

[17] M. Peter, W. Schindler, „A proposal for: Functionality classes for random number generators“, Version 3.0, September 10, 2024.

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

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

11 Annex

11.1 Development and Production Processes (life cycle)

11.1.1 General

360 This section contains additional information which is used for the refinements of the assurance requirements defined in 6.2.2.

361 The following section emphasises two different life cycles for the hardware platform. The first life cycle applies to hardware platforms where the Security IC Embedded Software is implemented in ROM. The second life cycle applies to hardware platforms where the Security IC Embedded Software is downloaded to the programmable NVM.

362 This PP is also applicable to products where both life cycles are combined. In this case, the hardware platform is customised by an initial Security IC Embedded Software which is supplemented by further Security IC Embedded Software parts that are downloaded to the programmable NVM. This may be applicable for Java Cards™.

11.1.2 Life Cycle Description

363 The Security IC Product life cycle is visualised in Figure 14 for products with customised ROM.

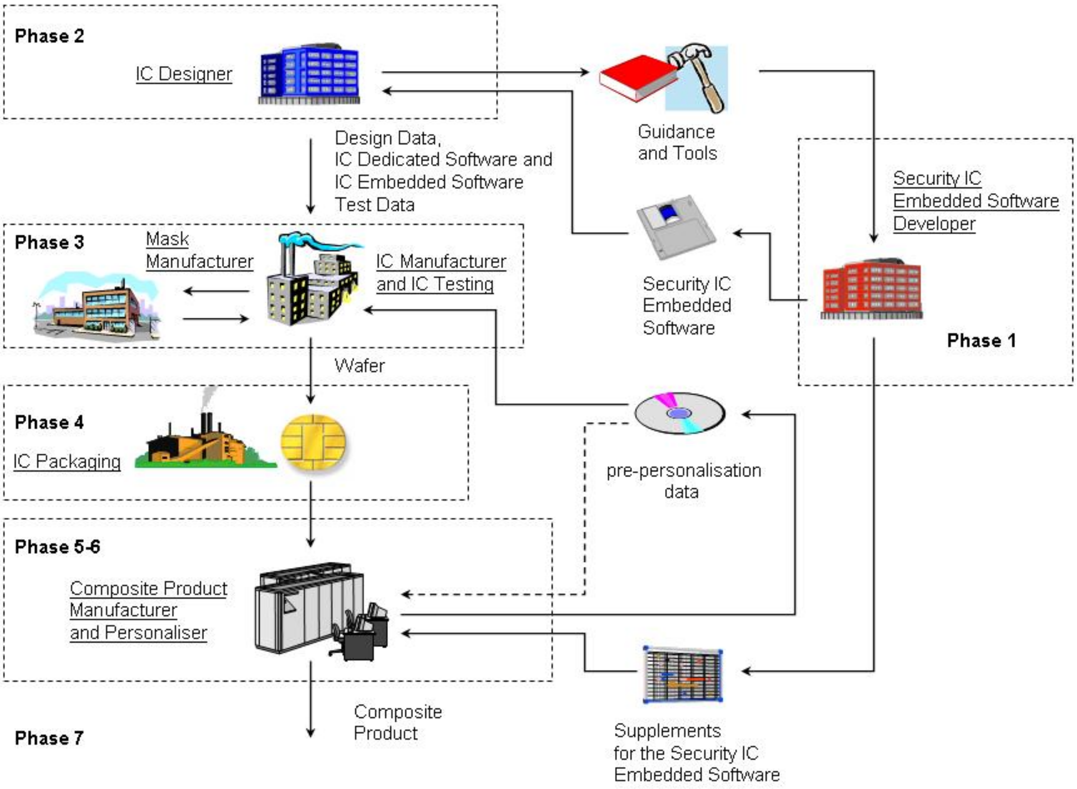


Figure 14: Security IC life cycle for Security IC Embedded Software implemented in ROM

364 The Security IC Product life cycle for products without customisation of the hardware platform is visualised in Figure 15. In this case the Security IC Embedded Software is stored in programmable NVM.

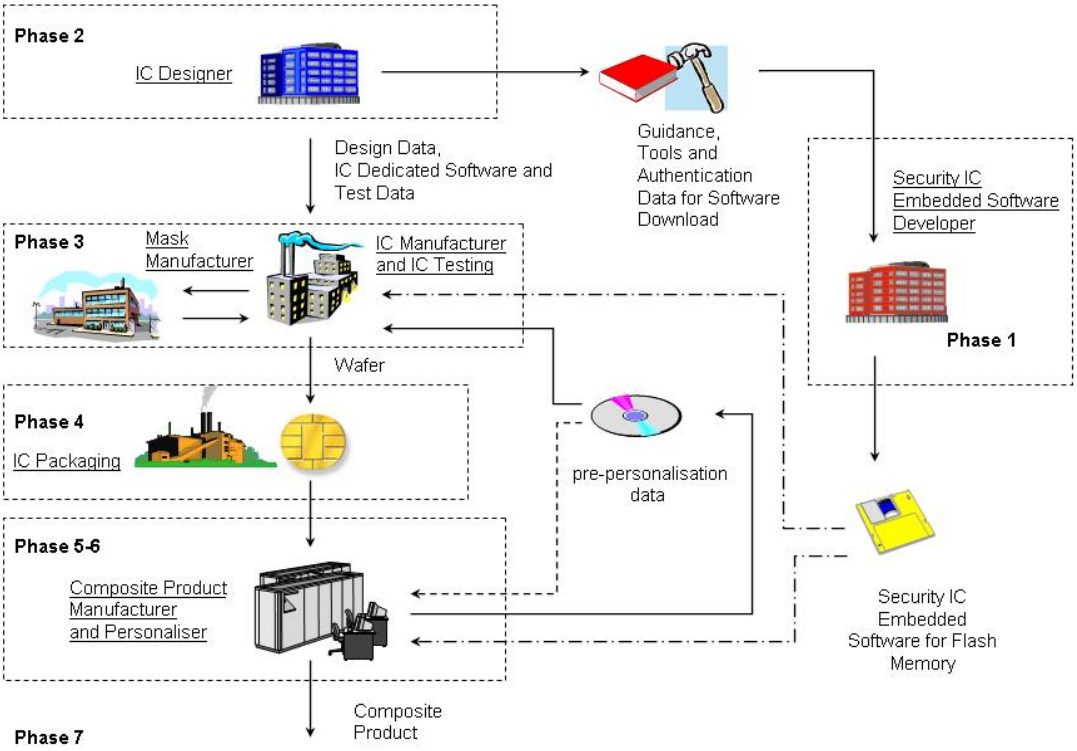


Figure 15: Security IC Life Cycle for Security IC Embedded Software loaded by the Security IC Dedicated Software into the programmable NVM

365 The Security IC Product life cycle is decomposed into seven phases where the following entities are involved. For the main differences between the two life cycles depicted above refer to the foot notes in the table.

Phase 1	Security IC Embedded Software Development	The **Security IC Embedded Software Developer** is in charge of • the Security IC Embedded Software development and • the specification of IC pre-personalisation requirements, though the actual data for IC pre-personalisation come from Phase 6 (or Phase 4 or 5)[47].

[47] For NVM-based products this includes also requirements for the secured download of the Security IC Embedded Software.

Phase 2	IC Development	The **IC Designer** - designs the IC, - develops the IC Dedicated Software, - provides information, software and tools to the Security IC Embedded Software Developer, and - receives the Security IC Embedded Software from the developer, through trusted delivery and verification procedures.[48] From the IC design, IC Dedicated Software and Security IC Embedded Software, the **IC Designer** - constructs the Security IC database, necessary for the IC photomask fabrication.
Phase 3	IC Manufacturing and Testing	The **IC Manufacturer** is responsible for - producing the IC through three main steps: IC manufacturing, IC testing, and IC pre-personalisation. The **IC Mask Manufacturer** - generates the photomasks for the IC manufacturing based upon an output from the Security IC database.

Application Note 47: If the Security IC Embedded Software is stored in ROM, the development of the software must be finished in Phase 1 and delivered to the TOE Manufacturer. If the Security IC Embedded Software is stored in programmable NVM, the TOE comprises a loader as part of the IC Dedicated Software and the Security IC Embedded Software can be downloaded. The download may be performed as a service provided by the IC Manufacturer or IC Packaging Manufacturer for the Composite Product Integrator before TOE Delivery, or by the Composite Product Integrator after the TOE Delivery. In the latter case, the delivery of the Security IC Embedded Software to the TOE Manufacturer is not required and Phase 1 can be performed in parallel to Phases 2 to 4.

Phase 4	IC Packaging	The **IC Packaging Manufacturer** is responsible for - the IC packaging and testing.

[48] This item is not required if the TOE is a Flash product. In this case the TOE Manufacturer must provide the information for the download of the Security IC Embedded Software.

Application Note 48: Phase 4 can be covered by the evaluation, refer to 1.3.5. The ST shall define the TOE Delivery point and indicate if Phase 4 is part of the evaluation.

Phase 5	Security IC Product Finishing Process	The **Composite Product Manufacturer** is responsible for • the Security IC Product finishing process and testing.
Phase 6	Security IC Personalisation	The **Personaliser** is responsible for • the Security IC personalisation and final tests.
Phase 7	Security IC End-usage	The **Security IC Issuer** is responsible for • the Security IC Product delivery to the Security IC End-consumer, and the end-of-life process.

366 If the TOE comprises programmable NVM the Security IC Embedded Software may be loaded onto the chip in Phases 3, 4, 5 or 6.

367 The relation between the semiconductor industry (TOE Manufacturer, refer to 1.3.5, in particular comprising the roles IC Designer, IC Manufacturer and Mask Manufacturer) and the other parties being involved in the Security IC development and production (especially the Security IC Embedded Software Developer) are visualised in Figure 16.

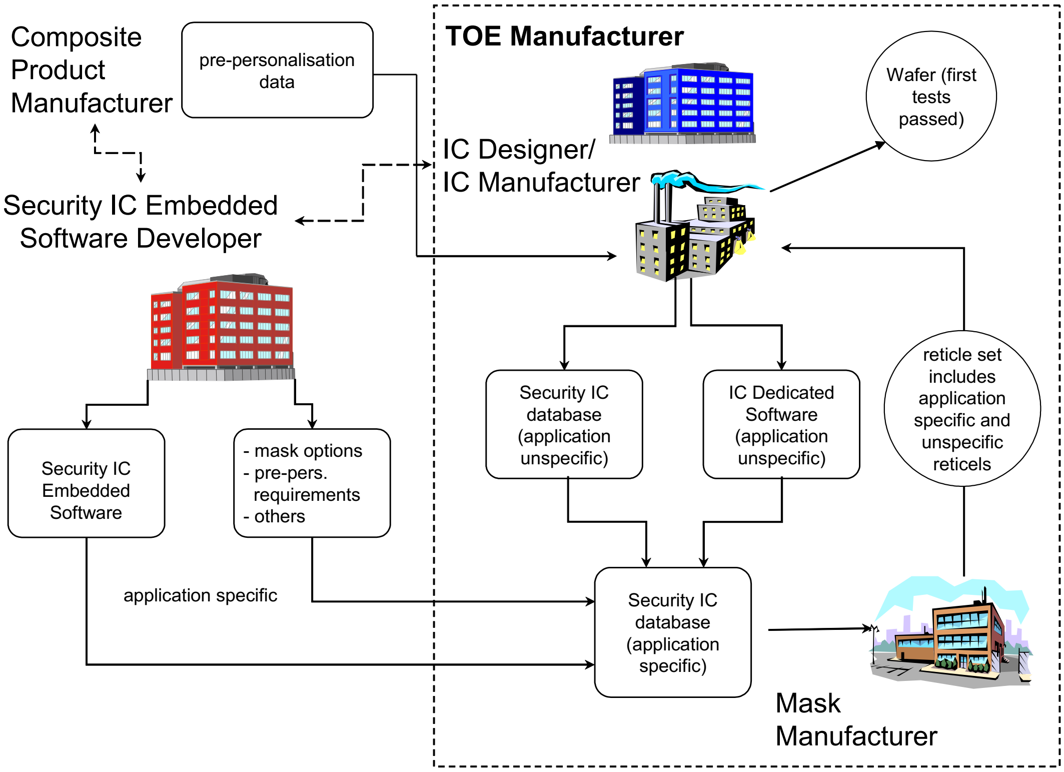

Composite Product Manufacturer
pre-personalisation data
Security IC Embedded Software Developer
TOE Manufacturer
IC Designer/ IC Manufacturer
Wafer (first tests passed)
Security IC database (application unspecific)
IC Dedicated Software (application unspecific)
reticle set includes application specific and unspecific reticels
Security IC Embedded Software
- mask options
- pre-pers. requirements
- others
application specific
Security IC database (application specific)
Mask Manufacturer

Figure 16: Development and Wafer Production including Testing in case of Embedded Software in ROM and NVM

368 For NVM-based products and similar TOEs, the design of the hardware platform is not customised, and the Security IC Embedded Software may not be delivered to the TOE Manufacturer. This is visualized in Figure 17. In this case, the Security IC Embedded Software is loaded in a later phase. To ensure the control of the software download, sufficient authentication mechanisms must be implemented by the IC Dedicated Support Software. Associated authentication data and/or keys must be exchanged between the TOE Manufacturer and the developer of the Security IC Embedded Software.

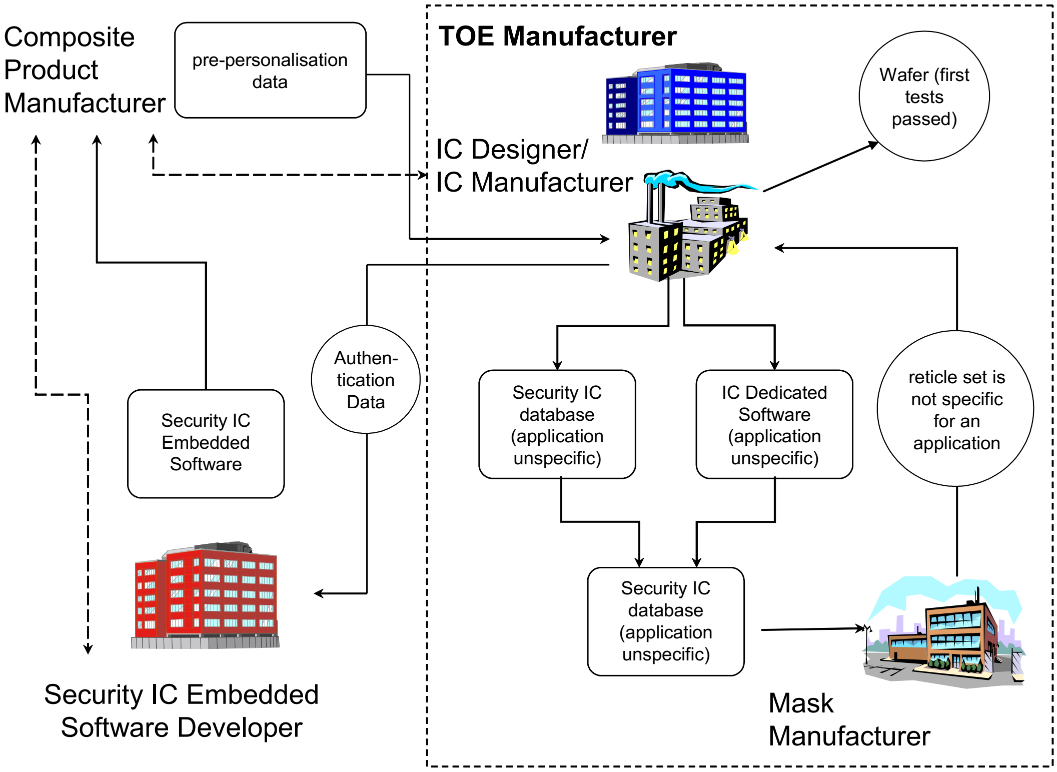

Composite Product Manufacturer
pre-personalisation data
TOE Manufacturer
IC Designer/ IC Manufacturer
Wafer (first tests passed)
Authen-tication Data
Security IC Embedded Software
Security IC database (application unspecific)
IC Dedicated Software (application unspecific)
reticle set is not specific for an application
Security IC database (application unspecific)
Security IC Embedded Software Developer
Mask Manufacturer

Figure 17: Development and Wafer Production including Testing in case of Embedded Software in programmable NVM only

369 The development process of the TOE starts with a process qualification. In parallel the concept of the TOE and the corresponding logical design is developed. The design uses standard library elements (circuitry and layout) which could be used for other (non-security) integrated circuits but may include full custom elements specially designed for the TOE as well. Some cells have parameters: For instance, the concrete layout of a ROM cell is determined by its contents which in turn is determined by the software or the data to be stored within.

370 All these cells not only differ in their logical or physical behaviour but also in their structure size which may range from very few elements such as simple gates up to physical units or sub-circuitry which may represent independent logical processing units. The physical cells (physical layout information is used) are placed on the chip area and then connected by wires (routing). Information about the physical layout of cells, about their position, about the shape of connecting wires and other process information define the physical layout of the chip.

371 These development steps are complex. Only the development of the logical design may have similarities with classic software development. However, technological constraints (such as timing) make this process more complicated and require, for instance, simulations which take technological and layout information into account. So, logical and physical design are developed in close relation.

372 The development of the information which defines the physical layout of an integrated circuit is a very complex matter. The photomasks or reticles that are required for wafer production are basically produced based upon this information. However, a bunch of technology-related parameters (possibly including parameters that depend on the wafer foundry) are considered in addition.

373 The photomasks or reticles are used to realise the integrated circuitry on/in a substrate. This again comprises tens of processes each affecting the result. Not only the layout principles but also the process information is proprietary to the IC Designer / IC Manufacturer. Each single chip (die or dice) is tested after production.

374 Development and production are based upon well-established processes of the manufacturer of the TOE. These processes are continuously improved mainly in order to increase yield and reliability.

375 During integrated circuit development and production lots of information and material is produced as summarised in 11.1.3. The evaluator must concentrate on the security critical assets and assess precisely their storage and handling. It is not sufficient to assess the company, arguing that personnel are trustworthy and exchange of information and material with external partners is properly controlled.

11.1.3 Description of Assets of the IC Designer/Manufacturer

376 The assets of the manufacturer of the TOE to be protected during development and production of the TOE are defined in paragraph 67 (page 20). Further explanatory text is given here.

377 The logical design data are those used to design the schematics of the chip (schematics or HDL sources and design documents). With the logical design data, the functionality of the chip can be understood. The logical design data can be regarded as being independent from the actual implementation (layout) though they contain the timing characteristics of some functional units (circuitry blocks).

378 The physical design data comprises all topographic information (three dimensional) about parts of the chip or the whole chip. Topographic information is the absolute or relative position, form, thickness, length and size of any structures realised on the chip surface. These structures are pads, connecting wires, isolation layers, vias, and implants.

379 The IC Dedicated Software, Security IC Embedded Software (if delivered to the IC Designer/Manufacturer), Initialisation Data and Pre-personalisation Data comprises the binaries and related documentation, and any data to be injected into the TOE before TOE Delivery. Source code may be required in specific cases.

Application Note 49: If the Security IC Embedded Software of the Composite Product is loaded by the Manufacturer of the Composite Product into the programmable NVM, the IC Designer/Manufacturer and the Photomasks Manufacturer may not need to know this Embedded Software. In this case the Pre-personalisation Data includes authentication data to control the access provided by the loader as part of the IC Dedicated Software for loading the Embedded Software.

380 The specific development aids comprise all tools especially developed to produce the product. One important example is the “ROM translator” which produces the physical memory content from the software binaries.

381 The test and characterisation related data comprise all information, which is used for testing including test results (pre-layout, post layout and product) and the characterisation of the final chip.

382 The material for software development support comprises all information and material given to the Security IC Embedded Software Developer to support the development of the Security IC Embedded Software.

383 The photomasks and products comprise the photomasks or reticles (usable and scrap) and chips (usable and scrap) in different forms.

384 The requirements of the CC assurance family ALC_DVS apply to all the above items. This entails the assessment of all the sites which are involved in the development and production of the product. Any exception request must be submitted to and approved by the certification body.

11.2 Guidance for FCS_RNG

385 This section provides examples of security requirements defined for random number generation in the BSI and NIAP certification schemes and how to perform the operations in the FCS_RNG.1. These examples are only informative.

11.2.1 BSI

386 The Bundesamt für Sicherheit in der Informationstechnik (BSI) published mandatory evaluation requirements for the German CC certification scheme [16]. These documents describe predefined classes PTG.2, PTG.3 and DRG.4 of random number generators (cf. [17]) which are appropriate for the TOE of this Protection Profile. We refer to [17] for examples of instantiations of the elements FCS_RNG.1.1 and FCS_RNG.1.2, including PTG.2 which is one of the most used classes.

11.2.2 NIAP

387 The National Institute of Standards and Technology (NIST) published the NIST SP 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators (June 2015) [18] and the NIST SP 800-90B Recommendation for the Entropy Sources Used for Random Bit Generation (January 2018) [19].

388 If the TOE implements a physical random number generator as entropy source compliant to [19] the ST author may define a SFR “Random Number Generation – ES (FCS_RNG.1/ES)” like this:

FCS_RNG.1/ES Random number generation

Hierarchical to: No other components.

Dependencies: No dependencies.

FCS_RNG.1.1/ES The TSF shall provide a *physical*[49] random number generator that implements:

(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*][50].

FCS_RNG.1.2/ES The TSF shall provide [selection: *bits, octets of bits, numbers [assignment: format of the numbers]*] that meet

(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]*] [51].

389 The clause (ES.3) may describe conditioning components implementing NIST approved or non-approved cryptographic functions, which are optional in [19]. 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 $0 \le \varepsilon \le 2^{-64}$.

390 If the TOE implements hybrid random number generator of the TOE complying to [18] seeded by a physical random number generator as entropy source described above the ST author may define a SFR “Random Number Generation – Hybrid deterministic RNG (FCS_RNG.1/HD)” like this:.

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*[52] random number generator that implements: [selection: *CTR_DRBG, Hash_DRBG, HMAC_DRBG*] as defined in NIST Special Publication 800-90A [18][53].

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

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

[51] ([assignment: *a defined quality metric*]

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

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

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*][54].

391 Refer to NIST Special Publication 800-90A [18] for details about the security capabilities and the security bits as quality metric of the random number output.

11.3 Examples of Attack Scenarios

392 This section provides additional information to facilitate the understanding of the threats defined in 3.2 and the different types of influence on and interactions with the Security IC which are shown in Figure 6. This does not constitute a comprehensive guidance for the evaluation.

393 A standard tool used for electrical measurement (and application of voltage and injection of current) is the needle probe workstation. Often appropriate contact areas must be prepared before using the methods described in the threat T.Phys-Manipulation. The actual measurement is done using standard tools such as voltmeters, oscilloscopes and signal analysers.

394 In addition, there are indirect methods for measurement which do not require a direct (metallic) contact. Examples are voltage contrast imaging and electron probe microscopy. These methods are also referred to as physical probing since the Security IC must be prepared before using the methods described in the threat T.Phys-Manipulation.

395 The Security IC carrier and therefore the surface of the integrated circuit constitutes the interface for these attacks.

396 The application of appropriate combinations of attack methods to reveal information (via a non-standard interface) are addressed by the threat T.Phys-Probing.

397 The malfunction of the TOE may cause some of its TSF to fail to be effective and this often propagates to the security functions or mechanisms of the Security IC Embedded Software. The most straightforward way to cause malfunction is to induce irregular operating conditions in amplitude, shape, timing, occurrence etc. on the ISO interface (for instance, glitches). Malfunction can be due to errors or premature ageing.

398 The attacker stimulates the ISO interface, e.g. power supply, the external clock, reset and/or I/O. The attacker may also consider other types of influence on the Security IC including attacking the surface of the integrated circuit. In this case, it might be required to manipulate the Security IC (refer to the threat T.Phys-Manipulation). In addition, the attacker needs to observe the behaviour of the Security IC and immediately take advantage of a possible malfunction. This requires having additional equipment such as a terminal and communication software, and possibly other depending on the application under attack.

399 The application of appropriate combinations of such methods to manipulate the Security IC Embedded Software (or the IC Dedicated Test Software) while being

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

executed (via a standard interface) are addressed by the threat T.Malfunction.

400 Specific sorts of malfunction are a means to reveal information about cryptographic keys or other critical data. Such methods are addressed by the threat T.Leak-Forced.

401 Standard tools used for the manipulation of circuitry are the Focused Ion Beam (FIB) and the laser cutter. The contents of programmable memories (such as NVM) may be modified for instance by manipulation of circuitry, by exposing cells to charged particle beams, by using electromagnetic waves or by electrical probing (application of voltage and injection of current).

402 The manipulation of the Security IC requires prior extensive reverse-engineering. The methods being applied are for instance optical inspection, voltage contrast imaging, image processing and pattern matching. In order to analyse the circuitry, the chip hardware must be removed from its carrier and then de-layered using appropriate methods such as wet etching, plasma etching or grinding.

403 The Security IC carrier and therefore the surface of the integrated circuit constitutes the interface for these attacks.

404 The application of appropriate combinations of methods to perform manipulation are addressed by the threat T.Phys-Manipulation.

405 When the Security IC processes user data of the Composite TOE and critical information about these data may be contained in signals which can be measured on the ISO contacts of the Security IC using standard tools such as voltmeters, oscilloscopes and signal analysers. The Security IC may also produce emanation which can be received using an antenna and analysed. For the analysis of the measured data specific tools (software) are required.

406 The interface for the attack is the ISO interface (contacts of the Security IC) but other interfaces may also be used.

407 The application of appropriate combinations of methods to reveal information without affecting the TOE’s operation or the TOE itself are addressed by the threat T.Leak-Inherent. Public known attack scenarios are for instance the Simple Power Analysis (SPA) and the Differential Power Analysis (DPA).

408 An attacker may also apply methods to cause the TOE to leak information. For instance, the attacker must in addition cause faults. The interface for the attack can be more complex in this case. The ISO interface (contacts of the Security IC), the Security IC itself and/or the surface of the integrated circuit may be used to cause faults (refer to the threat T.Malfunction). Physical manipulation is also possible (refer to the threat T.Phys-Manipulation).

409 The application of appropriate combinations of methods to reveal information (by affecting the TOE’s operation or manipulating the TOE itself) are addressed by the threat T.Leak-Forced, which is not related to attacks on cryptographic algorithms only. Public known attack scenarios are for instance the Differential Fault Analysis (DFA) and the Bellcore type of attacks.

410 In many cases, the evaluation of the TOE may not lead to definite results for the

products built using the TOE, and tests must be repeated or specifically designed with for the products embedding both the Security IC and the Security IC Embedded Software.

411 Test Features (including other non-application related functions) implemented in the TOE might be abused to disclose or manipulate user data and to bypass, deactivate, change or explore security features or functions of the TOE. Details depend on the capabilities of the Test Features provided by the IC Dedicated Test Software which are not specified here.

412 If the IC Dedicated Test Software offers commands via the ISO I/O interface, an attacker needs to communicate with the Security IC using a terminal and communication software. If other interfaces are used and/or if the usage of such commands is protected, it can be necessary to manipulate the TOE (refer to the threat T.Phys-Manipulation) and/or to circumvent authentication mechanisms. An attacker may also reveal information by physical probing (refer to the threat T.Phys-Probing) or analysing data (refer to the threats T.Leak-Inherent and T.Leak-Forced). If the TOE provides a command interface, this can be manipulated as described under the threat T.Malfunction and the software must not be affected by invalid inputs and other types of logical attacks specific for that software. Details depend on the way the Test Features are provided and protected by the TOE, which is not specified here.

413 The application of appropriate combinations of methods to reveal information or perform manipulation are addressed by the threat T.Abuse-Func.

11.4 Summary of Changes

414 This section provides an overview of the changes to the baseline PP [14].

415 This PP is aligned with CC:2022, Revision 1. Several SFRs that were formerly defined as extended requirements are now integrated as standard components. The updates include revisions to SFRs based on CC:2022 Part 2.

416 In addition, the three optional cryptographic packages “TDES,” “AES,” and “Hash Functions” defined in [14] have been replaced by a generic optional functional package “Cryptographic Services”, and a new optional functional package, “Address-based Access Control” has also been added.

417 Other structural and editorial updates include the organisation of the introduction, the presentation of the packages and annexes, the introduction of ALC_FLR.2, the update of the conformance claims, the references, the glossary and the abbreviations.

418 The changes are detailed in Table 11-1.

Table 11-1: Changes to the PP0084 v1.0

Change	Change Type	Description
Sponsors	Modification	The list of sponsors has been revised to include Thales and to remove Inside Secure.

Baseline reference	Modification	BSI-CC-PP-0084-2014 references the baseline Protection Profile BSI-PP-0035, whereas this PP v2.0 references BSI-CC-PP-0084-2014 as its baseline reference.
Introduction	Modification	The section has been restructured to facilitate the mapping with ASE_INT.1 content requirements
Incorporating new functional package	Addition	A new optional functional package, "Address-based Access Control," has been introduced. This package is derived from Addition #4: "Area-based Memory Access Control," as defined in the *Smartcard Integrated Circuit Platform Augmentations, version 1.00.*
Revising the functional packages for cryptographic services	Modification	The three cryptographic packages, i.e. "TDES," "AES," and "Hash Functions", defined in BSI-CC-PP-0084-2014 have been replaced by a general-purpose optional package "Cryptographic Services". This new package retains alignment with the original organisational security policy, objective, and SFRs, while introducing a generic formulation of the objective to accommodate various cryptographic algorithms.
CC Conformance Claim	Modification	This version of the PP claims conformance to CC:2022 Revision 1, instead of CC v3.1.
Functional Package Claim	Addition	A new sub-section has been introduced to specify the claimed functional packages clarifying that their inclusion is optional.
Assurance Package Claim	Addition	The assurance package claim has been augmented to include ALC_FLR.2 as an additional assurance component.
Conformance Claim Rationale	Addition	A new "Conformance Claim Rationale" section has been added to justify the inclusion of the claimed functional packages. This section demonstrates that the functional packages are consistent with the core PP and serve only to extend its functionality. It further explains that the SPDs, security objectives, and SFRs introduced by the functional packages neither contradict nor undermine those

		defined in the core PP.
Updates to SFRs in CC:2022 Part 2	Modification	SFR: FCS_RNG.1 The instantiation of FCS_RNG.1 in BSI-CC-PP-0084-2014 is identical to its definition in CC:2022 Revision 1. As the requirement is defined equivalently in both versions, it has been removed from the Extended Components section and is now solely included under the Security Functional Requirements for the TOE.
	Modification	SFR: FMT_LIM.1 The differences between the instantiation of FMT_LIM.1 in BSI-CC-PP-0084-2014 and its definition in CC:2022 Revision 1 are limited to terminology and have no impact. Consequently, the component has been removed from the Extended Components section and is included solely in the Security Functional Requirements for the TOE.
	Modification	SFR: FMT_LIM.2 The differences between the instantiation of FMT_LIM.2 in BSI-CC-PP-0084-2014 and its definition in CC:2022 Revision 1 are limited to terminology and have no impact. As a result, the component has been removed from the Extended Components section and is now included solely in the Security Functional Requirements for the TOE.
	Modification	SFR: FDP_SDC.1 The instantiation of FDP_SDC.1 in BSI-CC-PP-0084-2014 is equivalent to its definition in CC:2022 Revision 1. Accordingly, the component has been removed from the Extended Components section and is included solely in the Security Functional Requirements for the TOE.
Updates to SARs in CC:2022 Part 3	Modification	Different terminological and editorial updates have been applied to reflect the definitions in the Security Assurance Requirements. These changes are

		editorial and have no impact on the assurance content.
	Modification	SAR: AVA_VAN.5 The definition of AVA_VAN.5 has been updated to align with CC:2022. The core evaluation activities remain unchanged. However, new elements AVA_VAN.5.2D and AVA_VAN.5.2C require the developer to provide a list of third-party components included in the TOE or its delivery. Evaluator action AVA_VAN.5.2E has been extended to ensure that these components, as well as dependent IT products, are considered during vulnerability analysis.
	Addition	The rationale for ALC_FLR.2 (Flaw Remediation) has been added to justify its inclusion in the assurance requirements.
Updated SFRs in Package “Authentication of the Security IC”	Modification	SFR: FIA_API.1 The component FIA_API.1, previously defined as an extended component in the functional package “Authentication of the Security IC” in BSI-CC-PP-0084-2014, is now part of the standard components in CC:2022. The updated definition introduces assignments for the authenticated entity and its properties. Nevertheless, the underlying functionality remains consistent. Therefore, the component has been removed from the Extended Components section and is included solely in the Security Functional Requirements for the TOE.
Updated SFRs in Package Loader 1 “Loader dedicated for usage in secured environment only”	Modification	SFRs: FMT_LIM.1 and FMT_LIM.2 The SFRs have been updated in the same way as for the core PP.
Updated SFR in Package “Cryptographic Services”	Modification	SFR: FCS_CKM.6 FCS_CKM.4 used in the packages “TDES”, “AES” and “Hash Functions”, has been obsoleted in CC:2022 and replaced by FCS_CKM.6. The updated component allows the specification of cryptographic

		keys to be destroyed, along with the rationale for their destruction. The package “Cryptographic Services” includes FCS_CKM.6 and is in this sense more expressive than the packages defined in BSI-CC-PP-0084-2014.
Guidance for FCS_RNG	Modification	In BSI-CC-PP-0084-2014, this section provided informative examples of security requirements for RNG conformant with BSI and NIAP schemes. In this PP, the content related to the BSI scheme has been simplified and the corresponding document with examples is referenced.
References	Modification	References have been updated.
Glossary	Modification	The definition of the terms has been revised and corrected where necessary.
Abbreviations	Modification	The list of abbreviations has been revised to include previously omitted entries and remove those that are not used.