Supporting Document
Mandatory Technical Document

The Application of CC to Integrated Circuits

April 2006

Version 2.0
Revision 1

CCDB-2006-04-003

Foreword

This is a supporting document, intended to complement the Common Criteria version 2 and the associated Common Evaluation Methodology for Information Technology Security Evaluation.

Supporting documents may be “Guidance Documents”, that highlight specific approaches and application of the standard to areas where no mutual recognition of its application is required, and as such, are not of normative nature, or “Mandatory Technical Documents”, whose application is mandatory for evaluations whose scope is covered by that of the supporting document. The usage of the latter class is not only mandatory, but certificates issued as a result of their application are recognized under the CCRA.

Technical Editor: BSI

Document History
V2.0 April 2006 (revision by the JIWG)
V1.2, July 2002 (original CC-Supporting document)

General purpose:

The security properties of both hardware and software products can be certified in accordance with CC. To have a common understanding and to ensure that CC is used for hardware integrated circuits in a manner consistent with today’s state of the art hardware evaluations, the following chapters provide guidance on the individual aspects of the CC assurance work packages in addition to the Common Evaluation Methodology [CEM].

Field of special use: Smart cards and similar devices

Acknowledgments:

The governmental organisations listed below and organised within the Joint Interpretation Working Group contributed to the development of this version of this Common Criteria Supporting document.

France:	*Direction Centrale de la Sécurité des Systèmes d'Information*
Germany:	*Bundesamt für Sicherheit in der Informationstechnik*
Netherlands:	*Netherlands National Communications Security Agency*
United Kingdom:	*Communications-Electronics Security Group(CESG)*

Table of contents

1 Introduction

1 Today's increasing use of information technology has led to new methods of storage and processing of information. As a result of the packaging density on silicon, in the form of a microchip and improvements in performance, it is possible to process more and more information in parallel and at speed.

2 Complex microchips, which are able to process information, unfortunately introduce risks and dangers as well as huge advantages. Dependence on trouble-free functioning, as well as on the effectiveness of the protection measures, which have been carried out at the system and chip levels, has grown a great deal.

3 One must therefore be aware of the increased opportunities to test important information systems, including hardware against accepted criteria, in order to make the assurance of the security measures more transparent to the manufacturer, operator and user.

4 This publication has been reproduced in the Framework of the fundamental work for Certification. It should serve as a handbook for the application of CC to hardware components in respect of integrated circuits. This document will be of particular interest to manufacturers, evaluators and certifiers.

5 It has been developed and agreed within the European Joint Interpretation Working Group (JIWG) responsible for harmonising the application of CC between the European Schemes.

1.1 Objective

6 The security properties of both hardware and software products can be certified in accordance with CC. To have a common understanding and to ensure that CC is used for hardware integrated circuits in a manner consistent with today's state of the art hardware evaluations, the following chapters provide guidance on the individual aspects of the CC assurance work packages in addition to the Common Evaluation Methodology [CEM].

7 This guidance is applicable to the hardware of single ICs. It covers assurance levels as defined in CC, i.e. EAL1 to EAL5. These are the evaluation levels used for hardware integrated circuits today.

2 Assurance requirements for Integrated Circuits

2.1 Introduction

8 In applying CC to hardware components, two types of Target of Evaluation (TOE) may be considered:

- a TOE produced from a series of discrete parts on a printed circuit board or as a hybrid through different or several dice on one carrier.

- a TOE produced as an individual integrated circuit (IC).

9 The following guidance concerning the CC assurance aspects for a TOE is applicable to the hardware of single ICs.

10 In general, logical functionality in an IC can be implemented in simple PLD structures (Programmable Logic Device), FPGA structures (Field Programmable Gate Array), as an ASIC (Application Specific Integrated Circuit), or as well as a customer IC.

11 In respect of security applications, intelligent memory ICs with a hard-wired security logic (e.g. phonecards) or micro-controller based ICs in an ASIC design (e.g. electronic purse ICs) are used mainly for the elementary core components of security functionality.

12 Contemporary memory in ICs is based on EPROM or E^2PROM cells. This enables the non-volatile storage of data. Security logic can be utilised to implement identification and authentication, access control and internal IC sequence controls.

13 Micro-controller-based ICs offer the possibility of carrying out independently complex processes controlled by an IC operating system. Items, which also belong to the aforementioned functionality, comprise accountability functions and services such as encryption and digital signature, functionality that is implemented in hardware as well as software.

14 The mechanisms for the protection of software and operational data in various memories and the internal sequences are realised through the hardware of an IC (e.g. by means of certain technical measures and technological features) in order to support logical functionality.

15 An operating system of a micro-controller IC is contained (placed) in a ROM and/or an E^2PROM memory, and is protected from disclosure and modification during the operational phase by the technical or technological properties of the hardware. While technological properties are inherent in a TOE, technical properties depend on the design of the TOE.

2.2 CC Approach for Evaluation

16 The CC prescribe a variety of assurance activities, such as design analysis, vulnerability analysis, penetration testing, and examination of development environment. Whilst there are differences in terminology used, fundamentally the two sets of criteria are very similar in terms of specifying assurance requirements, and in their underlying philosophy.

17 A unique feature of the Common Criteria is the introduction of the Protection Profile concept. A Protection Profile can be characterized as a generic Security Target for a particular class of TOE (e.g. Smartcard ICs, as in the case of [PP-9806]), which defines a recognized standard to which conformance may be claimed. Whilst it is not mandatory to claim conformance to a Protection Profile, the availability of such standards provides the potential to reuse evaluated material in a Security Target, thereby considerably easing the process of producing this particular evaluation deliverable.

18

19 The following chapters provide guidance for hardware TOEs that have to be evaluated under Common Criteria (CC).

20 The evaluation of the IC comprises the following activities:

- Security Target,
- Development,
- Tests,
- Guidance,
- Configuration Management,
- Life-cycle support,
- Delivery and operation,
- Vulnerability assessment.

21 A Protection Profile (PP) may be defined, evaluated and certified in advance of a real TOE Evaluation and can be referenced within the Security Target of the real TOE Evaluation. In this case a TOE is conformant to a PP only if it is compliant with all parts of the PP.

22 The above mentioned activities correspond to certain assurance classes defined in the Common Criteria Part 3. The following table shows the Assurance Class / Assurance family breakdown and mapping.

Assurance Class	**Assurance Family**	**Abbreviated Name**
Class APE: Protection Profile Evaluation		APE[1]
Class ASE: Security Target Evaluation		ASE[2]
Class ADV:	Functional specification	ADV_FSP

[1] The assurance class APE is split into several components (see CC Part 3, chapter 4).

[2] The assurance class ASE is split into several components (see CC Part 3, chapter 5). The following discussion of the aspects of a Security Target follows these components.

Assurance Class	**Assurance Family**	**Abbrevi ated Name**
Development	High-level design	ADV_HLD
	Implementation representation	ADV_IMP
	TSF internals	ADV_INT
	Low-level design	ADV_LLD
	Representation correspond ence	ADV_RCR
	Security policy modelling	ADV_SPM
Class ATE: Tests	Coverage	ATE_COV
	Depth	ATE_DPT
	Functional tests	ATE_FUN
	Independent testing	ATE_IND
Class AGD: Guidance documents	Administrator guidance	AGD_ADM
	User guidance	AGD_USR
Class ACM: Configuration Management	CM automation	ACM_AUT
	CM capabilities	ACM_CAP
	CM scope	ACM_SCP
Class ALC: Life cycle Support	Development security	ALC_DVS
	Flaw remediation	ALC_FLR
	Life cycle definition	ALC_LCD
	Tools and techniques	ALC_TAT
Class ADO: Delivery and operation	Delivery	ADO_DEL
	Installation, generation and start-up	ADO_IGS
Class AVA: Vulnerability assessment	Covert channel analysis	AVA_CCA
	Misuse	AVA_MSU
	Strength of TOE security func tions	AVA_SOF
	Vulnerability analysis	AVA_VLA

Table 1 - Assurance family breakdown and mapping

23 CC assurance families are split up into certain hierarchical assurance components. These components are directed to predefined Evaluation Assurance Level (EAL) packages (EAL1 to EAL7). A combination of certain assurance components builds up one of the predefined EAL-packages as shown in CC Part 3, Annex B.

24 To perform a TOE evaluation, an EAL-Package can be used as predefined by CC (Part 3 conformant[3]). Besides this, augmentation[4] or extension[5] of an EAL package is possible (for details refer to CC, Part 1).

25 The following discussion gives guidance on how to use the assurance components of CC Part 3 assurance classes for hardware IC TOEs (e.g. a smartcard IC).

26 It is recommended to use one of the predefined EAL-Packages for a TOE Evaluation. In most cases, for an evaluation of a hardware IC (e.g. smartcard IC) an augmentation of the selected EAL package is necessary to fulfil specific objectives (e.g. for a high-level vulnerability assessment). If so, all dependency requirements as outlined within CC Part 3 have to be fulfilled. The following discussion will refer to augmentation aspects within the relevant paragraphs.

2.3 CC Protection Profile (Class APE)

27 Unlike a ST, which describes implementation oriented security; a PP describes abstract security requirements. For instance, a maximum rate of information leakage could be expressed in a PP, and a compliant ST could then state how a particular TOE approaches this requirement in terms of the split between hardware and software.

28 The majority of guidance applicable to a ST applies equally well to a PP (see para. 3.4) for example, definition of scope and boundary of TOE, environmental assumptions, threats, security objectives.

29 Various IC related PPs exist:

[PP-9806] covers IC hardware with only the most basic software. It has been developed by a community of semi-conductor manufacturers;

[PP-9810] which addresses security requirements for smartcard software;

[PP-9911] developed by Eurosmart which addresses security requirements for both smartcard software and hardware. This PP is build on [PP-9806].

[PP-SCSUG] is a PP, which presents a requirement for a complex set of IC hardware and software.

30 For actual PPs refer to the www.commoncriteriaportal.org on the Internet.

---

3 The CC define Part 3 conformant as:

4 Part 3 augmented is defined as:
A PP or TOE is Part 3 augmented if the assurance requirements are in the form of an EAL or assurance package, plus other assurance components in Part 3.

5 Part 3 extended is defined as:
A PP or TOE is Part 3 extended if the assurance requirements are in the form of an EAL associated with additional assurance requirements not in Part 3 or an assurance package that includes (or is entirely made up from) assurance requirements not in Part 3.

2.4 CC Security Target (Class ASE)

2.4.1 Objectives

31 The Security Target (ST) for a TOE is the basis for the evaluation and shall be agreed between sponsor, developer and evaluator. The audience for the ST is not confined to those responsible for the production of the TOE and its evaluation, but may also include those responsible for managing, marketing, purchasing, installing, configuring, operating and using the TOE.

32 Annex C of CC Part 1 describes in details the content and presentation requirements of the ST.

33 A Security Target comprises the following:

- a precise description of the TOE and its intended environment in terms of threats, any assumptions, organisational security policies and intended use.
- a description of the security objectives for the TOE and for the environment in order to determine whether the security objectives counter the identified threats, achieve the identified organisational security policies and adhere to the stated assumptions.
- any Protection Profile claims.
- a description of the security functional and assurance requirements of the TOE.
- a summary specification of the TOE security functions and the assurance measures.
- rationale giving justification for suitability and binding of defined functionality.

34 Within the following pages some observations concerning individual requirements are made.

2.4.2 Input

35 The developer shall provide the document ”Security Target”.

2.4.3 Requirements

TOE description

36 The Security Target shall include a precise description of the Target of Evaluation (TOE) in terms of hardware, software or firmware components. Reference to technological parameters is also important. The TOE description shall explicitly state the nature of any dedicated test software (either embedded software or software outside the integrated circuit).

37 The general security characteristics of the hardware shall also be described. A reference to the hardware datasheet would be appreciable. All possible configurations or intended use of the chip shall be documented.

38 The TOE needs to be clearly identified and separated from its technical and operational environment. The ST shall uniquely reference the TOE.

39 Since the hardware parts of an IC are both physically and functionally difficult to separate from one another without additional information, it is not possible to exclude

parts of the hardware from the TOE; it is therefore sensible to define the whole of the IC hardware as the TOE. In the case of certain parts of the IC being outside the TOE, a clear, logical and physical interface must exist. The inclusion of strongly hardware-oriented software/firmware in the TOE is appropriate. It would be sensible to look at an IC in its entirety and not only at the hardware or only at the software.

Security Environment

40 The security environment of the TOE comprises the operational environment after delivery as well as the technical environment in the different phases of the lifecycle of the TOE.

41 Therefore, a precise description of the TOE lifecycle is required or should be referenced. The boundaries of the TOE in terms of lifecycle shall be defined.

42 The Security Target shall explicitly state which phases are under the scope of the evaluation and which phases are excluded; the phases where the TOE is being developed and manufactured shall always be within evaluation scope.

43 In contrast with purely software TOEs, as in the cases noted here, this determination is only possible with precise knowledge of the manufacturing process. However, this determination also has a direct influence on the threat and attack scenarios in the operation of the TOE, which could be adopted in the context of the evaluation of the TOE. In the case of an IC TOE, the cut-off for evaluation could be after testing the IC as a die (at the earliest), or upon completion of packaging and associated testing. Optionally, the Security Target may include further phases of the IC such as micromodules assembly, prepersonnalisation and personalisation phases.

44 The CC explicitly requires that threats be characterized in terms of an identified threat agent, the asset at risk, and the attack. Thus it is necessary to define and enumerate all subjects in terms of roles and the assets for which specific protection either by the TOE or its environment is required and with reference to the lifecycle phases of the TOE.

45 Any assumptions placed on the environment, with which the TOE or its environment must comply, have to be identified.

46 Assumptions relating to the operation of software which is not part of the TOE are essential These may include:

- integrity protection software, e.g. for responding to sensors, or to watchdog timer interrupts);
- implementation of algorithms that are DPA-resistant;
- fault-handling software (e.g. protecting against inducing faults to enable a differential fault analysis attack on cryptographic keys).

47 Regarding the operational phase of the TOE, any assumption on the security aspects of the environment in which the TOE will be used or is intended to be used shall be described. Generally for a smartcard IC, no specific assumption for the TOE and its environment during the end-user phase (operational environment) has to be defined since this environment is a public one. However, if the TOE comprises only IC-hardware, there may be important security assumptions for the usage-phase of the TOE.

48 With reference to the lifecycle phases which are beyond the scope of the evaluation, there should be information about who is able to use the TOE after delivery and in what operational modes it is possible to use it. In this context, the actions of all personnel who come into contact with the TOE after delivery need to be examined. Therefore, specific assumptions about the behaviour of such personnel need to be defined and an identification of operational roles involved is necessary. Note: hardware may add a variety of roles that are IC-specific, e.g. there may be different approaches to personalization and enablement; such hardware-specific roles may need to be documented outside of the ST.

49 As an example for specific assets for a smartcard IC this may include:

- IC specific data including personalisation data and cryptographic keys,
- smartcard embedded software,
- IC dedicated software,
- IC specification, design, development tools and technology.
- Specific Application data like keys, authentication data or access control information.

50 A distinction between ***primary*** assets - such as information used by an user within the operational phase of the TOE (e.g. card holder) - and ***secondary*** assets which, if compromised, could be exploited to compromise a primary asset, might be useful. Secondary assets do not have any intrinsic value as such, but instead derive value from the primary assets. This distinction would allow a separation of high and low-level assets, which in turn will help to structure the statement of threats and thereby lead to a better understanding of the security objectives and security requirements to be met by the IC.

51 However, CC does not mandate that low-level or secondary assets are identified in order to drive the selection of security objectives and requirements. For example, integrity protection SFRs can be included simply to help achieve a security objective for protection of a high-level or primary asset - with the security requirements rationale explaining the purpose of such SFRs.

52 Assumed threats to the assets shall be described. CC requires that threats defined in the ST, are not ***directed*** at the identified security objectives, but rather are ***addressed*** by the security objectives. It should also be noted that the CC explicitly requires that threats be characterized in terms of an identified threat agent, the asset at risk, and the attack (describing such aspects as attack methods employed, vulnerabilities exploited, and opportunity).

53 It shall be a description in terms of damages to the assets rather than attack paths, which could not be completed in the ST. The threats shall refer to the TOE environment life cycle phases (development, production or operation) or the TOE itself. They could be described in terms of:

- unauthorized disclosure of assets,

- unauthorized use of assets,
- unauthorized modification of assets.

54 To be able to understand the threats defined for the environment of the TOE in certain phases of the lifecycle, the TOE's development and production environment shall be described.

55 Beside logical functionalities, technical and technological properties can be attacked during the operational phase of the TOE, too. The corresponding threats to the defined assets can therefore be formulated (e.g. selection of objects also by means of physical attacks, operation of the TOE outside specific parameters, such as voltage, frequency and temperature).

56 With respect to determining specific threats, it should be noted that attacks on ICs during production processes, and in particular during test phases, are possible and can be defined for the relevant life-cycle phases. They also can become apparent during the operational phase in the form of vulnerabilities (c.f. Development Security, Vulnerability Assessment), evaluated within the relevant assurance class ALC and AVA, for example.

57 Specifications of organizational security policies depend essentially on the applications in which the TOE is incorporated. Generally speaking, for a pure hardware smartcard IC evaluation, no specific organizational security policy has to be defined.

Security Objectives

58 The CC requires that security objectives be specified within the ST for both the TOE and the environment that are necessary to counter the threats and uphold the identified assumptions and organizational security policies.

59 The objectives may apply to different life-cycle phases of the TOE construction such as objectives on the TOE development, objectives on the TOE manufacturing or on the TOE delivery process. These objectives will mostly be fulfilled by the evaluation of certain assurance aspects for the TOE such as Configuration management (ACM), Delivery and operation (ADO) and Life cycle support (ALC). Additionally, objectives apply to the TOE environment in the operational phase of the TOE after delivery.

60 The objectives shall be clearly stated to permit a clear mapping back to the relevant threats. They could be derived from the following:

- tamper resistance of the TOE,
- protection from cloning,
- protection of sensitive memory,
- protection from information leaking.

61 Therefore technical and technological properties of the IC shall be addressed.

62 Additionally, there may be a need for specific security objectives on the environment to ensure that assumptions concerning dependencies on software are upheld.

Security requirements

63 Security requirements shall be defined within the ST using the functional and assurance components specified in Part 2 and Part 3 of the Common Criteria. In some cases, if predefined functional components of CC part 2 are not applicable, new IC-specific components might be defined within a ST.

64 It is required that the security functional requirements (SFR) and assurance requirements (SAR) on the TOE are needed to meet the identified security objectives for the TOE.

65 Additionally, security objectives for the environment within certain lifecycle phases might be satisfied by measures for the environment of the TOE evaluated by the assurance requirements for the development process of the TOE. (e.g. development security).

SAR:

66 The minimum Strength of Function (SOF) rating should also be specified. In most cases it would be sensible to claim a SOF-High rating for a smartcard IC. Nevertheless, according to CC this claim is applicable only to permutational and statistical functionalities (see AVA_SOF below for more information).

67 The required level of attack potential for the vulnerability analysis is expressed in CC requirements by selection of a certain AVA_VLA-assurance component. This component defines the baseline for the protection of the TOE in terms of attack potential against which the vulnerability analysis of the TOE will be judged.

68 With reference to CEM, Annex A.8, the evaluators independent vulnerability analysis goes beyond the vulnerabilities identified by the developer. The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low (for AVA_VLA.2), moderate (for AVA_VLA.3) or high (for AVA_VLA.4) attack potential.

69 By way of example, to ensure resistance against high attack potential for vulnerability analysis at EAL4, the AVA-class has to be augmented with AVA_VLA.4. Additionally, the components which AVA_VLA.4 depends on have to be required within the SAR. For more information on attack potential refer to [AP-SC].

SFR:

70 Some guidance on the use of such components in a PP or ST for an IC might be helpful, although there are now a number of PPs in existence for ICs which can be usefully referenced (indeed, a need to comply with such PPs may pre-determine which functional components must be used in the ST).

71 Possible CC Part 2 Security Functional Requirements (SFRs) for an IC comprise for example components from the following classes: FIA, FMT, FCS, FDP, FPT.

72 It is important that the selected functional components be tailored to the extent necessary to demonstrate that the TOE security objectives are met. This applies to both PPs and STs (but especially the former, where operations on functional components can be left uncompleted, thus resulting in SFRs, which are too generic.

73 A further issue is the CC requirement that SFRs are actually *testable*. For guidelines on testing refer to ATE below.

74 The FPT_PHP components are used to express requirements for protection of the TOE from physical tampering attacks and require the TOE to implement functions to respond to these attacks - whether by:

- providing the capability to detect an attack (FPT_PHP.1); or
- detecting and providing notification of an attack (FPT_PHP.2); or
- responding automatically to resist an attack (FPT_PHP.3).

75 The selection of these components might be adapted to the situation in place. For example, responding automatically to resist an attack (FPT_PHP.3) may be refined as assuming that there might be an attack at any time and therefore providing countermeasures at any time because the TOE´s technical properties may not be able to detect the attack but are activated in place permanently. For example, permanent protection against Differential Power Analysis is required, ensuring that the TSP could not be violated at any time.

76 Because there is no assurance requirement for hardware diagnostic procedures at any CC EAL, such requirements may be expressed as Security *Functional* Requirements using the FPT_AMT.1 or FPT_TST.1 functional componentsTrusted recovery is also not an assurance requirement of EAL5. This may also be expressed as a Security *Functional* Requirement using FPT_RCV componentsFrom the need to integrate software with hardware, specific requirements may arise to be able to reflect the use of the hardware TOE in the software design. Because the relevant issues need to be identified and addressed in the evaluation deliverables, requirements must be stated in the ST. For example, incrementing a retry count before making a PIN check would be an example of this, as would the inclusion of transaction protection structures and states (to deal with breaks in communication with the IC). In particular, the model of computing will need to be dealt with (hence requirements such as those to protect integrity are likely to be considered).

77 In the evaluation of a composite TOE (IC hardware plus software: operating system, application software and IC dedicated software) it may be applicable to select functional components for an information flow control policy through the use of functional components from the FDP_IFC and FDP_IFF families. These components apply to certain parts of the software which are part of the TOE (for example such requirements are placed on the OS and hence on the integrated platform comprising IC and OS; in [PP-9806] such components are applied to IC dedicated software.

78 A ST or PP may modify selected CC part 2 components to be more meaningful to smartcards, (however, it should be noted that modified requirements like this need to be proven in practice) such as:

- A deviation from audit data generation component, FAU_GEN.1 to exclude the requirement for date and time in the audit record. However, this requirement is only achievable if an externally trusted time source exists and trust can be preserved in the record.

- A refinement of FPT_TST.1, self-processing, to include card blocking functions.

Operations on requirements:

79 The Security Target shall explicitly perform all the operations (assignment, iteration, selection, and refinement) of the security requirements. These operations may concern both functional security requirements as well as assurance security requirements. As a minimum, all the operations of assignment and selection of functional security requirements shall be performed.

80 For each selection, the Security Target author shall select the appropriate item in the selection list. For each assignment, the Security Target author shall specify the appropriate item. Any guidance could be found in CC Part 2 annexes. Since the specification of security requirements is independent of the way it is implemented (hardware, firmware or software), the level of specification shall be the same as the level of functional requirements.

81 For ICs, it is important that security aspects of design and implementation, which are not necessarily functional in nature, be addressed by the evaluation.

TOE summary specification (TSS)

82 The TOE summary specification provides a high-level definition of the security functions claimed to meet the TOE functional requirements (SFR) and the assurance measures to meet the TOE assurance requirements (SAR) as defined in the ST.

Security Functions:

83 The description of the IT security functions could be identical to the security functions specifies within the TOE functional specification (FSP) but the tendency is for them not to be as detailed.

84 Irrespective of whether the TOE comprises software and/or hardware, the specification of the TOE security functions within the Security Target should provide specification of the security properties of the TOE at an abstract level. This specification can be independent of the method of implementation.

85 In this sense, the concept of Security Functions (SF) has been extended beyond the traditional logical and functional meaning to contain technological or technical properties implemented in an IC. The following types of SFs should be considered:

- SFs in the form of logical functionality[6] (e.g. authentication logic) which are directed against logic attacks; they counter threats directly.

- SFs which, in addition to being logical functionality, comprise technical and technological properties, too. These functions can be implemented by a combination of passive structure with active logic (e.g. monitoring of supply voltage by means of integrated CMOS technology parameters in certain areas of the IC).

[6] Meant are: functionalities realized in hardware, but comparable to software functionalities.

- SFs in the sense of non-functional technical properties which make an attack more difficult (e.g. security which has purposely been built in). Memory circuits and function elements of the hardware are protected from disclosure and modification with the help of such technical properties. These technical properties are the result of measures employed in the development/ production process. SFRs in this category comprise bus obscurity (to guard against spoofing), protective layering and coating (passivation of the metallisation layer) and use of advanced process design.

86 All of the above types of SF should be taken into consideration during evaluation of an IC.

Assurance Measures:

87 The assurance measures could be defined by reference to the evaluation deliverables or by a general intent to adopt appropriate measures to meet the requirements. There can be assurance measures for certain phases of the life cycle in place.

88 The SOF ratings associated with specific IT security functions (implemented by permutational or probabilistic mechanisms) should also be defined.

PP claims

89 The Security Target shall explicitly claim compliance with any Protection Profile if applicable.

90 The protection profile [PP-9806] ”Smartcard Integrated Circuit”, which has been developed by a community of semi-conductor manufacturers could be referenced in the Security Target. Other Protection Profiles could also be referenced in the Security Target such as:

- [PP-9810] ”Smartcard Embedded Software Version 1.2” which addresses security requirements for the smartcard software ;

- [PP-9911] ”Smartcard Integrated Circuit with Embedded Software” developed by Eurosmart which addresses security requirements for both smartcard software and hardware. This PP is build on the PP/9806.

91 For actual PPs refer to the www.commoncriteriaportal.org on the Internet.

92 It shall be noted that claims of partial compliance to a PP is not permissible under the Common Criteria.

93 In case of any PP compliance, the Security Target does not need to repeat statements of security requirements included in the PP that are unmodified for the Security Target. Nevertheless, it could be easier to have an independent document.

94 If, however, the PP includes uncompleted operations, which is the case for PP/9806, refinement operations are the responsibility of the Security Target.

Rationale

95 The security target is fundamental to set up an effectiveness view since it lists the intended use of the IC, the operational environment, the assumed threats, objectives,

functional and assurance requirements and the SFs and assurance measures as discussed above.

96 CC requires a rationale which demonstrates that a TOE conformant with the ST will effectively address all relevant aspects of the 'security problem' defined by the *Statement of TOE Security Environment*. The ST rationale presents the analysis in a step-wise manner:

- firstly, the security objectives for the TOE and its environment must be shown to be suitable to counter the identified threats (transposed by certain attack scenarios) and uphold all identified policy needs and assumptions. If applicable, scenarios of physical attacks on the hardware can be of significance. Assumptions made relating to the software operation are essential.

- secondly, the security requirements must be shown to be suitable to satisfy the TOE security objectives, and be mutually supportive and provide an integrated and effective whole (binding). Therefore, the analysis should consider combinations of SFR where some of them may be logical and others technological and technical requirements. The analysis should also consider relations between assurance requirements and objectives for the environment of certain phases of the lifecycle like development and production phases.

- thirdly, the IT security functions and assurance measures must be shown to meet the security requirements. Therefore, the analyses should consider the combination of logical, technological and technical SFs as well as assurance measures so far as they are defined within the Security Target. Technical peripheral conditions of the specification, such as temperature, voltage and frequency should be considered if specified within the functional requirements.

97 Binding of hardware- and firmware-functionalities is to be taken into consideration, depending on the scope of the TOE. For an IC, the analysis would need to take any assumptions into consideration made relating to the software operation that are essential to meeting the Security Target.

98 Detailed aspects addressing the issue of indirect attacks against the IC (e.g. bypassing or contradicting the SEFs) should be part of the vulnerability assessment (see class AVA).

99 In the evaluation of a composite TOE, the analysis should discuss the interrelationships between the software and hardware parts of the TOE to demonstrate that they are mutually supportive in helping to meet the Security Target for the composite TOE. This will not only involve discussion of dependencies of the IC on the software such as those listed above, but also the software dependencies on the hardware, including tamper-resistance aspects.

100 Analysis is mostly done by providing mappings in combination with informal arguments of the mapped items and explanations on relations between certain items.

101 No further specific guidance is needed for application to ICs.

2.5 Development (Class ADV)

102 The assurance class ADV defines requirements for the stepwise refinement of the TOE Security Functions (TSF) from the TOE summary specification in the ST down to the actual implementation. Each of the resulting TSF representations provide information to help the evaluator determine whether the functional requirements of the TOE have been met.

103 The technical description of the TOE is always accompanied by mapping technical components of the TOE to security functionality.

2.5.1 Functional specification (ADV_FSP)

2.5.1.1 Objectives

104 The functional specification describes the TSF, and must be a complete and accurate instantiation of the TOE security functional requirements as defined in the ST. It refines the TOE summary specification from the ST.

105 The functional specification also details the external interface to the TOE. Users of the TOE are expected to interact with the TSF through this interface.

2.5.1.2 Input

106 Regardless of EAL level, the developer shall provide the functional specification.

107 The external interfaces are usually described in the data sheet for the IC. The ISO standard (7816) is of relevance in most cases. In addition, the die description shall be given.

2.5.1.3 Requirements

108 The functional specification usually uses developers' terminology. The level of detail required for the specification has to be correlated to the coverage of functional tests (ATA_COV) and to the external interface description within the guidance documents (AGD_ADM / AGD_USR).

109 Specification of functional details of the TOE security functions has to be provided within the Functional Specification. More detailed representation levels map these functional details to the defined subsystems or modules.

110 External interfaces of an IC can be:

- interfaces to the hardware parts of the IC which are not part of the TOE (provided that the TOE is appropriately separated), or

- interfaces to the software/firmware which is stored on the IC and is not part of the TOE, but which runs on the IC hardware under consideration (e.g. the triggering of an interrupt via hardware, test software interfaces), or

- logical interfaces (e.g. instruction set, special function register specification, memory map) and physical interfaces of the IC (IC contacts with serial I/O and supply, die description), which guarantee the connection to the outside

world within the operational environment and the operating system/ application programming environment.

111 Simply observing external interfaces from the point of view of their logical behaviour is unlikely to be sufficient. Externally adjustable operational parameters and their limits should also be investigated because direct attacks or vulnerabilities may result.

2.5.2 High-Level Design (ADV_HLD)

2.5.2.1 Objectives

112 The objective of these requirements is to determine whether the high-level design is sufficient to satisfy the functional requirements of the Security Target. It provides a description of the TOE Security Functions in terms of major structural units with functional coherence, provides a description of the interfaces to these structural units, and is a correct realisation of the functional specification.

113 This phase of the development process is essential to define the major components of the TOE (subsystems), the basic structure of the TOE, its external interfaces and its separation between major subsystems.

2.5.2.2 Input

114 High-level design information shall be made available at all EALs from EAL2.

2.5.2.3 Requirements

115 The high-level design is a top level design specification. Complete databook documentation of the chip may be considered to support these requirements. A block diagram, which originates in the design and conception phase, as well as an informal description, can be an integral part of the High Level Design.

Basic structure of the TOE in terms of subsystems and their interfaces:

116 Typically, the documentation needed for this set of requirements may be described as a mapping of the major architectural components to the physical devices performing specific functions (e.g. CPU, RAM, ROM, Bus and I/O elements).

117 In many cases, components which represent the general structure of an IC TOE can be definite logical units; they are possibly even implemented as a physical unit on the IC. Examples comprise: memory, data/address bus–memory interface, arithmetic block, contact interface, watchdog timer, sensors with analysis logic, controls for the voltage supply, logic blocks for access controls or authentication for memory ICs with security logic, a micro controller block on micro controller ICs.

118 Since the security properties of an IC TOE can consist of logical functionality as well as technical and technological properties, it is necessary to document the general structure of the architectural components as well as to explain the technical and technological structure (general layout rules of the physical design: technology, number of layers, bus routing) because of their importance to the hardware security properties. A protective layer could, for example, be seen as a component of the general structure of the physical composition of the TOE.

119 The developer's choice of subsystem definition, and of the grouping of TSF within each subsystem, are an important consideration in making the high-level design useful in understanding the TOE's intended operation. The number of subsystems, the choice of grouping of functions within subsystems together with the description of their purpose has to be appropriate and sufficient for the evaluator to gain a high-level understanding of how the functionality of the TSF is provided.

120 Additionally, it would be reasonable to expect complexity to be deliberately increased at the lower levels of IC representation.

121 Since EAL3 does not require a component from the ADV_LLD family but requires the depth of functional test at the level of the high-level design (ATE_DPT.1), the description of all interfaces to the subsystems is required to specify the tests.

Hardware and firmware required by the TOE

122 Since the TOE itself consists of hardware, further items of hardware may be required for its operation. It is noteworthy that an external voltage and timing supply, or a defined data interface can be necessary for the operation of the TOE. If it is the case that not all of the IC hardware is to be included in the TOE (e.g. there may be hardware parts on the IC which are needed for the TOE's subsequent operation) they should not be overlooked by evaluators in case vulnerabilities may occur.

Functionality of the supporting protection mechanisms

123 If supporting external protection mechanisms are available, a clear distinction between TOE internal mechanisms and external mechanisms is necessary. While doing so, dependencies within the hardware or possibly through the firmware which is involved should be taken into consideration. Only TOE internal mechanisms are subject to evaluation requirements.

124 A check sum algorithm could be implemented in firmware which may not be within the scope of the TOE but the result of the check used by the TOE. An exact description of the dependencies and the behaviour of the interface are necessary.

The separation of the TOE into TSP enforcing and other subsystems (from ADV_HLD.2 on)

125 At ADV_HLD.2 and above, the separation of TSP-enforcing subsystems from other subsystems of the TOE or from subsystems of the intended environment should be examined carefully. Mapping of the TOE security functional requirements to physical subsystems may not be easy to do (e.g. which subsystem really does process the security functions, the CPU or a CPU in conjunction with its associated memory and bus?). This is because within the IC itself there are strong dependencies between various physical components at the implementation level, which complicate an effective separation in the sense of the criteria. Consequently, it is mostly necessary and may be easier for some hardware TOEs to classify all of the subsystems of an IC TOE at the level of the high-level design as TSP-enforcing.

126 If subsystems are separated into TSP-enforcing and others, the required justification about the clearness and effectiveness of the separation (at ADV_HLD.4 and above) should be based on logical and physical dependencies. A maximal independence of

subsystems within an IC TOE could be possible if there were no or only minimal physical overlaps and logical dependencies between the individual subsystems and the interfaces are clearly defined.

127 For example Test-ROM firmware could be classified as ”other subsystem” because it is deactivated in the operational phase of the TOE. Another example, depending on the specific security functionality of a TOE, may be a standard peripheral unit, e.g. a timer, if separation can be shown.

Semi-formal notation (E4)

128 At EAL5, the high-level design is required to be semiformal. The semi-formal description of the architecture can take the form of block circuit diagrams or hardware description language documents (HDL – hardware description language). In many cases, however, a hardware description language would first be used at the Detailed Design level.

129 Meaningful graphical representations of the technical or technological properties as part of the SEFs of the TOE may be considered equivalent to semi-formal representation.

130 The informal documentation of the technical and technological properties, as well as their integration into the structure and the realization of SEFs, is necessary.

Evidence of how the Security Functionality is provided by each subsystem of the TSF

131 Assigning the Security Functionality to subsystems is especially difficult, since individual subsystems not only provide the realization of a single Security Function and very strong interactions and dependencies between subsystems exist. Mapping of the TOE security functional requirements of the ST via the security functions of the Functional Specification to physical subsystems may not be easy to do as mentioned above. For this reason, a description of the functional flow of Security Functions to defined subsystems is of particular significance.

132 The required evidence is combined with the evidence of correspondence between the TSF representations (ADV_RCR). Therefore each security function identified in the Functional Specification (FSP) shall be mapped onto a TSF subsystem described in a high-level design.

133 The SFRs of the ST expressing security properties might be realized and traced to technological properties of the TOE as described in the design.

134 The required justification (at ADV_HLD.4 and above) that the TSF mechanisms are sufficient to implement the security functions identified in the high-level design should be based on logical and physical aspects of the design.

2.5.3 Low-Level Design (ADV_LLD)

2.5.3.1 Objectives

135 The objective of these requirements is to determine whether the low-level design is sufficient to satisfy the functional requirements of the Security Target. It has to be a correct and effective refinement of the high-level design, and needs to provide sufficient information to support other evaluation activities.

2.5.3.2 Input

136 Low-level design information shall be made available by the developer at EAL4 to EAL7.

2.5.3.3 Requirements

137 If the IC design is managed through a classical process of hardware drawings, the development process depends essentially on the technologies used (specific method and tools) and can be described by refining high-level design into a sufficient level of detail to implement the TOE.

138 If the IC design is managed through a hardware description language (HDL), the low-level design is similar to those used in classical software development.

139 The construction plans and functional descriptions, which result from the use of an HDL tool and a CAD tool, can be used directly during the construction of the detailed design, but needs only to be presented fully from EAL4.

140 If necessary, the evaluator needs to be able to use the tools provided by the manufacturer for evaluation.

141 The design elements necessary for the construction of the TOE are:

- logical plans (which for example consist of analogue cells, standard cells, gates, transistors and diodes) or corresponding HDL-representations in order to realise individual functionality as well as SEFs,

- specification of the physical design which describes the requirements for the organisation of the physical components (e.g. module placement, layer order, routing specifications).

142 The technical and technological structure of the TOE is to be refined, in order to make an effectiveness analysis possible during the examination of physical attacks on the TOE in the context of the assumed threats.

Description of the TOE Security Functions in terms of modules, their purpose and interrelationships:

143 Modules are refined subsystem specified in the high-level design. In the case of an IC, the actual logic and layout plans will have been derived from modules, whereby the separation of the modules has to fulfil the testing requirements. This means that the interfaces of the modules and the modules itself need to be testable (from EAL5 because of ATE_DPT.2 requirements). Supplementary to this, it should be noted that modules at ADV_LLD.1 level at EAL4 (ATE_DPT.1 used) serve simply to refine the structure of the TSF.

144 Interfaces between modules must be described especially carefully, since there are strong dependencies between them in an IC. Functionality, which runs in parallel, should be considered with the description of the interfaces.

145 The timing of the module interfaces should be described if they are accessible from the outside (e.g. pads) for tests.

Semi-formal notation:

146 From ADV_LLD.2 on (EAL6) a semi-formal notation of the LLD is required and may be made available in the following forms: refined block circuit diagrams, flowcharts, conditional diagrams, truth tables, documents using a hardware description language (HDL). Modules must be indicated and described.

147 Meaningful graphical representations of the technical or technological properties as part of the TSF of the TOE as well as layouts may be considered equivalent to semi-formal representation.

148 At EAL5 the low-level design is not required to be semiformal, but for a low level design of an IC the state of the art design process uses semiformal methods anyway. So it might be easier for the developer to use semiformal notations for evaluation out of his normal design process even at EAL5 (ADV_LLD.1).

Realisation of the TSP enforcing functions:

149 CC requires descriptions of how the modules provide the security functions, and of the interrelationships between them. Additionally it is required to describe how each TSP-enforcing function is provided.

150 The specification of functional details of the TSF has to be provided already within the Functional Specification (see ADV_FSP). The Low-Level Design maps these functional details to the modules defined and describes the functional flow (e.g. by flow charts) and the behaviour of the modules.

151 If security properties are to be used for the realization of SEFs, which arise as a result, of certain design or layout requirements or because of demands on technology, they must be described with sufficient precision in connection with affected modules or interfaces. This is of particular significance to the subsequent vulnerability assessment.

152 Logical functionalities of an IC can be described as similar to mechanisms of a software TOE.

153 If technical and technological properties are part of a specified TOE security function, the effect of the technical and technological properties in the operation of the TOE need to be provided in the sense of a description of the purpose and behaviour of a module.

154 SFRs of the ST expressing security properties might be realized and traced to technological properties of the TOE as described in the design.

155 The required evidence is combined with the evidence of correspondence between the TSF representations (ADV_RCR). Therefore each TSF module identified in the low-level design shall be mapped onto a TSF subsystem identified in the high-level design (HLD).

The separation of the TOE into TSP enforcing and other modules

156 As outlined above for the ADV_HLD family, it is mostly necessary and may be easier for some hardware TOEs to classify all of the modules of an IC TOE even at the level of the low-level design as TSP-enforcing. Otherwise, if modules are separated into TSP-enforcing and others, justification should be provided. Justification should be based on logical and physical dependencies.

2.5.4 Implementation Representation (ADV_IMP)

2.5.4.1 Objectives

157 The objective of these requirements is to determine whether the implementation representation is sufficient to satisfy the functional requirements of the Security Target and is a correct realisation of the low-level design.

158 It is the least abstract representation of the TSF and captures the detailed internal workings of the TSF in terms of source code, hardware drawings, etc., as applicable.

2.5.4.2 Input

159 The implementation representation shall be made available by the developer at all levels from EAL4.

2.5.4.3 Requirements

160 The TOE implementation representation corresponds to the following:

- hardware drawings for analogic blocks;
- HDL statements for all synthesised components;
- if applicable, source code for all dedicated/embedded software;
- physical design information like layout and mask plans describing the implementation of the physical components, if required for analysing or performing specific attack scenarios.

161 Links between HDL documentation and analogic blocks shall be described and interfaces documented.

162 Implementation information and layout shall be made available to the evaluator if required. In respect of implementation understanding, the compiling chain from HDL to the IC, shall be documented (ALC_TAT.2).

163 An analysis of the implementation representation is performed by the evaluator with two objectives:

- analyse the correctness of the implementation representation and the low-level design (traceability of the security functions). A correspondence between low-level design and schematics/layout shall be provided for this purpose.
- understand the TOE implementation in order to specify potential vulnerabilities and attack paths.

164 Layout plans (physical design) describe the organization of the physical components and the signal routing with regard to the process masks and determine the metallisation masks.

165 Mask plans are necessary for the technological process. The mask plans need only be shown in certain cases if they are necessary for follow-up analyses like vulnerability analyses.

166 Layouts are required to check the correctness of the implementation of technical and technological properties. The layout indicates the ease with which physical attacks can be mounted (for example, the accessibility of the metallisation layer).

167 The technical and technological structure of the TOE is to be refined, in order to make an effectiveness analysis possible during the examination of physical attacks on the TOE (e.g. layout information about specific cells might be necessary if they are subject to certain attacks like FIB).

168 According to CC, the implementation level shall define the TSF at a level of detail such that the TSF can be generated without further design decisions.

169 The required description of relationships between all portions of the implementation (from ADV_IMP.2) shall include justification of relations between functional modules and the technical and technological structures of the TOE as well as between hardware and firmware parts of the TOE.

170 To support the correspondence analysis between low-level design and implementation (ADV_RCR), layout information as well as HDL-deliverables should be used. Additionally, a mapping between modules of the low-level design and the implementation level deliverables is necessary to show correspondence.

171 To show the completeness and accurateness of the implementation of the SFRs of the ST a mapping between ST requirements and implementation details is required. SFRs of the ST expressing security properties might be realized and traced to technical and technological properties of the TOE with respect to certain areas in the layout of the IC.

172 Adjustment of production process parameters will be determined at this stage via consideration of technology specific characteristics and the means of CAD tools. (See also para. 3.10.)

173 At EAL4 (ADV_IMP.1) the implementation representation is sampled by the evaluators. Under CC, however, a PP/ST author can direct the sampling towards areas of specific interest, through application of the refinement operation to the ADV_IMP.1 assurance requirement (e.g. requiring the selected subset of the implementation representation to include at least the subset of the physical structure of the TOE related to such things as interconnects and data bus layout, fuse locations, and physical structure including shielding layers and packaging).

174 For example, a FPT_TST.1 (self-tests at start-up and periodically thereafter) requirement may be achieved by environmental sensors, which in turn are represented by security logic and hardware schematics. In this case, operational envelope conditions would need to be traced through the various levels of FPT_TST.1 representation and test evidence.

2.5.5 Representation Correspondence Refinement (ADV_RCR)

2.5.5.1 Objectives

175 The objective of these requirements is to determine whether the developer has correctly and completely implemented the requirements of the Security Target's TOE summary specification, through to the least abstract TSF representation that is provided. The least abstract TSF representation is the functional specification at EAL1, the high-level

design at EAL2 and EAL3 and the low-level design and implementation representation from EAL4 to EAL7.

176 For each representation, correspondences between higher and lower representations of the TSF clearly identify the traceability from one TSF representation to the next.

2.5.5.2 Input

177 The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

2.5.5.3 Requirements

178 Depending on the notation of representations under correspondence, analysis itself has to be either informal, semiformal or formal as shown in the following table.

	EAL1	EAL2	EAL3	EAL4	EAL5	EAL6	EAL7
TSS to FSP	Informal	Informal	Informal	Informal	Informal	Informal	Informal
FSP to HLD	-	Informal	Informal	Informal	Semifor mal	Semifor mal	Formal
HLDto LLD	-	-	-	Informal	Informal	Semifor mal	Semifor mal
LLD to IMP	-	-	-	Informal	Informal	Semifor mal	Semifor mal

Table 2 - Required notation for the RCR analysis

179 It should be noted that design, test evidence and manuals will have to be cross-referenced to fulfil the TOE requirement for correspondence evidence between different representations of the target of evaluation, depending on the target assurance level.

180 Informal and semiformal correspondence analysis can be done by comparing statements, mapping items by using tables, or using detailed references. The analysis shall cover all details of the TSF. This includes technical and technological properties specified for a TSF as well as functional aspects. From EAL4 (ADV_LLD.1) modules as defined e.g. within the HDL code or in logic plans need to be traced down to the layout.

181 Graphical or programming tools used in the developer‘s IC design process may support the analysis from HLD via LLD to IMP. In this case the evaluator may use the developer’s tools to check the refinement.

2.5.6 Security policy modelling (ADV_SPM)

2.5.6.1 Objectives

182 The objective of these requirements is to determine whether the security policy model clearly and consistently describes the rules and characteristics of the TOE Security Policy.

183 Security policy models are structured representations of security policies of the TSP, and are used to provide increased assurance that the functional specification corresponds to the security policies of the TSP, and ultimately to the TOE security functional requirements. This is achieved via correspondence mappings between the

functional specification, the security policy model, and the security policies that are modelled.

2.5.6.2 Input

184 The developer shall provide a TOE Security Policy (TSP) Model at EAL4 to EAL7.

2.5.6.3 Requirements

185 At EAL4 the TSP model shall be expressed informally (ADV_SPM.1). At EAL5 to EAL7, it shall be expressed using appropriate formal languages such as Z or VSE-SL (ADV_SPM.3).

186 The TOE Security Policy Model shall include:

- Definition of policy rules to be enforced by the TOE, typically covering access control or information flow control policies.

- Definition of TOE characteristics, e.g. subjects (active entities) and objects (passive entities).

187 It is likely that a CC ST would contain sufficient information to satisfy the ADV_SPM.1 requirements; thus there is no specific interpretation of CC required.

188 The formal security model is a formal description of the security policy. It is utilized at an abstract level independently from the TOE implementation in hardware or software.

189 As a guidance on the formality requirements for technical and technological properties (e.g. difficulty of reverse engineering, or operation out of envelope) it may help to model the protection of the IC against unauthorized disclosure of assets, unauthorized use of assets, and unauthorized modification of assets in terms of barriers in combination with the security states of the IC during the different life cycle phases.

190 The CC does not require that *all* security functions be formally modelled, only those functions or policies that *can* be modelled.

2.5.7 TSF internals (ADV_INT)

2.5.7.1 Objectives

191 The TSF internals requirements specify the requisite internal structuring of the TSF in terms of modularity, layering, minimization of the complexity of policy enforcing mechanisms, and minimizing the amount of non-TSP enforcing functionality within the TSF, thus resulting in a TSF that is simple enough to be analysed.

2.5.7.2 Input

192 The developer shall provide an architectural description. The developer has to meet certain design requirements at EAL5 to EAL7.

2.5.7.3 Requirements

193 CC, part 3, 10.4 states: The architectural description is at a similar level of abstraction to the low-level design, in that it is concerned with the modules of the TSF. Whereas the low-level design describes the design of the modules of the TSF, the purpose of the

architectural description is to provide evidence of modularity, layering, and minimisation of complexity of the TSF, as applicable. Both the low-level design and the implementation representation are required to be in compliance with the architectural description, to provide assurance that these TSF representations possess the required modularity, layering, and minimisation of complexity.

194 In particular, a rationale how the chosen structure provides for modularity, layering, and minimisation of complexity should be based on logical and physical dependency.

195 A maximal independence of modules within an IC TOE could be possible if there were no or only minimal physical overlaps and logical dependencies between the individual modules and the interfaces are clearly defined.

196 However, in smart card IC design, obscurity objectives are of importance on the implementation level to increase the barrier for physical attacks (e.g. random module placement, glue logic). This may contradict the ADV_INT objectives. Nevertheless, in this case design information must be provided clearly structured and detailed to support the evaluation tasks.

2.6 Tests (Class ATE)

197 The assurance class ATE states testing requirements that demonstrate that the TSF satisfies the TOE security functional requirements. The CC distinguish four assurance families within this class: Test coverage (from EAL2), test depth (from EAL3), functional tests (from EAL2) and independent tests (from EAL1).

2.6.1 Coverage (ATE_COV)

2.6.1.1 Objectives

198 The objective of these requirements is to determine whether the testing (as documented) is sufficient to establish that the TSF has been systematically tested against the functional specification. Coverage deals with the completeness of the functional tests performed by the developer on the TOE.

2.6.1.2 Input

199 The developer shall provide evidence (at ATE_COV.1, EAL2) / an analysis (from ATE_COV.2, EAL3) of test coverage. This analysis may be part of the test documentation itself or supplied separately.

2.6.1.3 Requirements

200 The test coverage analysis shall consider the mapping between tests (characterization and production tests) and the TSF as described in the functional specification (security functions). The coverage analysis shall show that all properties of the security functions are covered.

201 The analysis has to show and justify whether the TOE has been comprehensively tested. Complete coverage of security functions and external interfaces is required at EAL3 and higher (ATE_COV.2). From ATE_COV.3 (EAL6) the analysis has to show that all external interfaces have to be completely tested. For an IC, this can mean for example that the complete instruction set of the CPU with all parameters be covered.

202 Regarding test coverage, attention must be paid to the inclusion of security functions which result from design or technology. Therefore, evidence has to be provided that technical and technological properties specified in the FSP are covered by tests or other appropriate activities (e.g. layout, mask and chip inspections).

2.6.2 Depth (ATE_DPT)

2.6.2.1 Objectives

203 The objective of these requirement is to determine the depth of testing. Depth analysis deals with the level of detail to which the developer tests the TOE. Testing of security functions is based upon increasing depth of information derived from analysis of the TSF representations, e.g. whether the developer has tested the TSF against its high-level design at EAL3.

2.6.2.2 Input

204 The developer shall provide an analysis (from ATE_DPT.1, EAL3) of test depth. This analysis may be part of the test documentation itself or supplied separately.

2.6.2.3 Requirements

205 The depth of testing analysis which is provided for these requirements shall consider the mapping between tests (characterization and production tests) and internal structures of the TSF. Depending on the level of evaluation, this is done at:

- the high-level design level (ATE_DPT.1, at EAL3 - EAL4),
- the high-level design and low-level design level (ATE_DPT.2, at EAL5 - EAL6) or
- at the high-level design, low-level design and implementation level (ATE_DPT.3, at EAL7).

206 This means that

- ATE_DPT.1 gives evidence on the level of TSF subsystems and their interfaces (external and internal),
- ATE_DPT.2 gives additional evidence on the level of TSF modules and their interfaces. Whilst [CEM] does not address the EAL5 requirements, by extrapolation it is likely that all modules and their interfaces would need to be covered in order to satisfy the ATE_DPT.2 requirements. The application of this testing requirement will depend critically on how the term ‘module’ is used for an IC (see comments on ADV_LLD above).

207 All deliverables provided at a certain level (e.g. block diagrams, HDL code, layout documents) should be used for examination of test depth.

208 For ATE_DPT.2, appropriate depth of testing is achieved when all instructions and branches of the whole logical plan vs. HDL source code, which belong to the TSP enforcing modules, have been tested.

209 The correctness of the implementation (integration) and the test coverage must also be proven after production (see ATE_FUN). The test vectors must be chosen appropriately, so that they cover the requirements. Analyses should be performed in this way.

210 The justification for test depth on the implementation level can be done with respect to one of the following items:

- The developer can, if possible, show that he has toggled each junction of a module during testing.

- If, according to the logical plan, the modules or parts of them can only be tested in parallel, the developer must show that all junctions have been toggled at least once via the underlying test vectors.

- If the developer has not taken testability rules into consideration, or the testing of some modules is not immediately possible (the testing of a timer

  over 24 hours), the circuit cannot be considered 100% tested. In this case, test coverage is achieved if the developer can show that all junctions were achieved via the test vectors and that there are no conditions which compromise security.

211 If using EAL4 for an IC evaluation, augmentation of EAL4 by the component ATE_DPT.2 might be sensible to get higher assurance, because the low-level design has been provided anyway and in case of testing technical and technological properties a low-level design view of tests is sensible.

2.6.3 Functional tests (ATE_FUN)

2.6.3.1 Objectives

212 The objective of these requirements is to determine whether the developer's functional testing demonstrates that all security functions perform as specified.

2.6.3.2 Input

213 The developer shall test the TSF and document the results. The developer shall provide test documentation. For the conduct of tests, IC data sheets are of particular importance.

2.6.3.3 Requirements

214 The developer test documentation is required to give details of test plans, goals, procedures and results (actual and expected). Because ATE_FUN.1 is used at EAL2 to EAL5, the quantity of information that must be presented will vary in accordance with the use of ATE_COV and ATE_DPT.

215 Tests of individual components of the TOE, or the control of certain technical or technological properties, could only possibly be implemented at a certain time during the manufacturing process or only in test mode, since the respective physical components can be neither logically nor physically accessed after the end of the production of the TOE. This should be considered during test planning and be appropriately documented.

Test plan:

216 The test plan has to show the objective of the tests, which are to give evidence for the correctness of the logic by means of simulation using the HDL tool and to test the correctness of the implementation. Since a test is a type of quality control, after simulation it must be proven whether the implementation has been successful. Individual tests on the finished IC must show that the implementation of the security functions and mechanisms is correct, and that the timing requirements are fulfilled. During testing specified functionality or during module testing, binding of modules is of particular note, especially if parallel functionality exists.

217 Typically, the two main steps in testing a hardware TOE are:

- the ”TOE prototype” tests and
- the acceptance tests performed on each TOE at the end of the production phase.

218 "TOE prototype" tests as characterisation tests that can be considered to provide evidence for the correct implementation of security enforcing functions. Timing should be considered when testing Hardware TOEs. Tests can also be implemented at the design level with the help of HDL tools (without delays, with estimated loads/ delays and post-layout as appropriate) or in form of special security tests after production using e.g. specific software residing in the TOE (test software within the ROM and being part of the TOE or application test software in the EEPROM not part of the TOE).

219 Acceptance tests have to confirm and verify the correct operation of the TOE and the components of which it is constructed during its manufacture. Therefore, the evaluators shall check the developer’s manufacturing process that it has appropriate acceptance tests implemented. The acceptance testing during production is usually performed using specific hardware mechanisms and commands implemented in the test software on the chip.

220 The different test environments used for evaluation shall be described within the test plan, e.g.

- ”TOE prototype” tests:
  characterisation test environment,
  design simulation environment during development,
  security testing environment and
- acceptance testing environment during production.

221 Equipment, which is necessary for a test case, must be specified exactly with all adjustments. This also includes the precise identification of the test libraries for simulation as well as the driver program for the test equipment.

222 In order to perform or to check the results of characterisation and acceptance tests where specialist test equipment is essential, the evaluator may have to witness and verify the tests rather than personally perform them. This is normally done through a visit to the IC designer/manufacturer.

223 If the developer would like to do security testing without simulation by means of the HDL tools, all tests must be carried out in real time in order to give evidence that the implementation be correct.

224 The library of test programs provided by the developer shall contain test programs and tools to enable all tests covered by the test documentation to be repeatable (required for ATE_IND). This may include driver software, among other things, with its associated equipment (tester) required for the testing of the chip. This is also necessary for repeating tests. Other tools that have been used, such as the logic analyser, oscilloscope, debugger, operating system etc. also need to be stated.

225 A test plan determines the framework of the test cases. In the test plan, the exact specification and scope of the test cases, as well as the documentation describing all input and environment parameters of the IC, are of great importance. These parameters are partially given in data sheets. Therefore the data sheet must be an integral part of the test documentation.

226 The test cases can vary greatly with analogue and digital circuits.

227 The test plan should cover all configurations of the IC if specified for the operational environment, e.g. different security states of the IC like test mode and user modes depending on the life cycle phases under evaluation.

228 Apart from the functional tests under standard conditions, tests (if necessary real time tests) under defined stress conditions (temperature, frequency, voltage, EPROM cycle tests etc.) are also to be planned, since such conditions could arise during the operation of the TOE (comparable with extreme situations for software TOEs, which could lead to run-time errors).

229 If during operation of the TOE external HW or SW functionality be included dynamically in the functional flow, then the relationship of the external components to the level of the external interface should be tested.

230 A mapping shall be given between the test cases and security functions and subsystems, modules or interfaces under test depending on test coverage and depth.

231 Test parameters must be taken into consideration in the test planning. They could be for example:

- test frequencies with minimum and maximum limits
- voltage supply corresponding to the data sheet
- test temperatures
- test vectors for the selection of the test areas in the IC

Test procedure:

232 Test procedure descriptions have to identify the behaviour of the security feature under test and have to give sufficient instructions to reproduce initial test conditions and to reproduce the stimulation of security features under test to observe their behaviour. For hardware tests the initial programming of registers and memories of the chip is of

importance. The order of initialization or stimulation steps in the timely order is of importance, too (start-sequence and the execution flow).

233 The test procedure is the instigation of the test plan using the planned test vectors. In the event that a logical part or memory area should be locked, this must be explained or described, together with the circumstances for locking.

234 If fusing is required for a security function of the ST or its refinement in the FSP, the methodology of fusing needs to be described in detail. This concerns specifically the deactivation of test hardware or the transition from the test- or initialization mode to the operational mode. Fusing specification can also be part of the assurance class Delivery and Operation (ADO). In any case the information provided in this respect has to be sufficient for a rating of the resistance against defusing attacks within the context of the effectiveness testing.

235 If security functionality is implemented using technical and technological properties, it may be sensible, to prove the evidence of a characteristic not only by functional testing, but to verify the presence of a special technical or technological property by analysing the chip (e.g. by optical inspection).

236 This indicates that non-functional (technical or technological) properties may not be testable by functional tests but for example by extensive design checks on the implementation level (hardware drawings and masks) or optical, e-beam or x-ray checks on the IC. On the implementation level, it may also be sensible to implement automatic checks on the digital layout files to ensure that, for example, bus lines with certain signal names in the VHDL code of the Low Level Design are scrambled over different layers or certain areas of a mask as required. Further assurance is gained by abundant penetration tests against non-functional properties.

Test results:

237 The way test results are presented by the hardware must be described (e.g. written to a register or certain memory area, sent via an external interface line).

238 The test results, which will be obtained on special test equipment, must be presented in a form that can be analysed (analogue tests, timing tests).

239 For test results concerning functionality which run in parallel, the assignment of the results to specific subsystems, modules or SFs is of significance. The dependencies of the results of the tests resulting from parallel processing functionality should be explained.

2.6.4 Independent testing (ATE_IND)

2.6.4.1 Objectives

240 The objective of these requirements is to determine whether the TOE behaves as specified and to gain confidence in the developer's test results by independently testing a subset of the TSF and by performing a sample of the developer's tests by a party other than the developer (e.g. a third party).

241 This family adds value by the introduction of tests that are not part of the developer's tests.

2.6.4.2 Input

242 The developer shall provide the TOE (from ATE_IND.1) and an equivalent set of resources (from ATE_IND.2). The evaluator shall provide test documentation.

2.6.4.3 Requirements

243 The equivalent set of resources the developer has to provide from ATE_IND.2 may include a separate sample of chips from production, a separate set of test vectors and a separate set of test data necessary for testing.

244 The evaluator shall provide test documentation. Requirements for test documentation are comparable to those for the developer's test documentation in terms of a test plan, procedures, and expected and actual results.

245 From the expertise point of view, the evaluator must be able to repeat the developer's tests and perform additional tests. For the conduct of tests, the evaluator needs test vectors, which determine the course of tests. If required, the evaluator must be in the position to use the tools for evaluation applied by the manufacturer. In many cases, owing to tool availability, this will only be possible in the development laboratory or during production by the manufacturer. In these cases, it is sufficient for the evaluator to witness the tests at the manufacturer's site.

246 Apart from the functional tests under standard conditions, tests (if necessary real time tests) under defined stress conditions (temperature, frequency, voltage, EPROM cycle tests etc.) are also to be performed by the evaluator, since such conditions could arise during the operation of the TOE and may not be extensively tested by the developer.

247 The additional evaluator tests must be performed at least at the level required by ATE_DPT.

248 The evaluators must also perform additional tests on a completed IC (final part), because:

- Errors could be introduced by technology and may not be detected by logic tests (cf. the ageing process in chapter 3.3.4),

- The scattering of security-enforcing and security-relevant parameters cannot be tested via simulation. Such scattering can only be carried out by means of testing several ICs. In order to do this, the evaluator has to select an appropriate sample or rely on the results of the manufacturer's quality tests. For example, the scattering of a mistake in digitalisation can only be detected if several ICs are tested, as assembly of basic components can lead to a timing deviation.

2.7 Configuration Management (Class ACM)

249 Configuration management (CM) helps to ensure that the integrity of the TOE is preserved by requiring discipline and control throughout its development and maintenance in the processes of refinement and modification of the TOE and other related information. CM prevents unauthorised modifications, additions, or deletions to the TOE, thus providing assurance that the TOE and documentation used for evaluation are the ones prepared for distribution.

2.7.1 Configuration Management capabilities (ACM_CAP)

2.7.1.1 Objectives

250 Configuration management capabilities define the characteristics of the configuration management system.

2.7.1.2 Input

251 The developer shall provide CM documentation including a configuration list of the TOE (from ACM_CAP.2 / EAL2). From ACM_CAP.3 / EAL3 the CM documentation shall include a CM plan. From ACM_CAP.4 / EAL4 the CM documentation shall include an acceptance plan.

2.7.1.3 Requirements

252 The configuration control system and the acceptance procedures should be considered during the whole development and production process of the TOE. They must be in a position also to control the construction plans and hardware parts, in addition to all relevant data files for all development steps. If applicable, various development and production sites are also to be included in this.

253 The evaluator should ensure that the TOE contains a unique reference such that it is possible to distinguish between different versions of the TOE. The TOE may provide a method by which it can be easily identified. For hardware TOEs this may be a part number physically stamped on the TOE. Furthermore, each TOE mask layer may be physically identifiable by use of any kind of identifiers.

254 However, in certain cases it can be necessary, in order to make an attack more difficult, to mark the ICs (or the chips held within the ICs) with non-visible logos or IDs. In such cases, the manufacturer must, however, find a suitably hidden possibility for the label on the TOE, such as for example, in a non-deletable area of memory with access for authorized users only.

261 The configuration list consists of all TOE items maintained by the CM system. From EAL3, these items are specified in ACM_SCP.

255 The evaluators workunit at ACM_CAP.4.11C/EAL4 to examine the TOE generation procedures has the objective to get evidence about the effectiveness of the configuration control system with regard to the various versions and changes to the TOE. It has to be shown that the configuration control system supports the generation process to help reduce the probability of human error. Thus, the generation process makes use of the appropriate design tools (HDL / CAD tools). This should be described.

256 In the case of an IC TOE comprised of hardware and software (e.g. Test-ROM software, operating system, smart card application software depending on the specific scope of a TOE) there is likely to be a distinction between the hardware and software configuration control. There is an additional requirement to bring together the right hardware-software pair, which means that the right mask must be used. This in turn means that the configuration control system must be able to administrate hardware-software pairs comprising a TOE. Because all masks are created from mask data files it has to be ensured that masks are created from their correct software image.

257 Depending on the scope of the TOE, bringing together the right hardware-software pair can partly be an aspect of the TOE configuration management or of the delivery procedures (see ADO-DEL).

258 It is not possible to relate this workunit for generation of the TOE directly to the technological process of a customer specific IC, because the process cannot be repeated for the benefit of the evaluator. In this case, the evaluator must, however, audit the configuration control in the technological process in order to guarantee that the correct masks, which belong to a particular version of the TOE, are used and organizational measures are effective in the process.

259 In the case of programmable standard ICs (PLD, FPGA), in which the hardware configuration is programmed via firmware, the workunit may be supported by programming a new IC. The evaluator then conducts comparison tests of the functions of the newly produced IC with the original TOE (to be equated with a 'file compare' for a re-built software TOE).

2.7.2 Configuration Management scope (ACM_SCP)

2.7.2.1 Objectives

260 The objective of these requirements is to determine the scope of the configuration management of the TOE by indicating the TOE items that need to be controlled.

2.7.2.2 Input

261 The developer shall provide CM documentation.

2.7.2.3 Requirements

262 Configuration Management shall concern both the TOE and its documentation.

263 For example, at ACM_SCP.2/EAL4 level, as a minimum, the developer performs configuration management on the TOE implementation representation (hardware schematics/layouts), design documentation, tests, user and administrator guidance, the configuration management documentation and security flaws.

264 Configuration management shall be in place for IC design (schematics, layout) as well as IC proprietary dedicated software (source code, documentation). All source files necessary for the generation of the TOE as well as the identification of the set of masks necessary for the manufacturing of the TOE have to be included. For masks, this includes unique mask identifier, as well as the version number or revision number of each layer.

265 Because the configuration control system must be able to administrate hardware-software pairs as part of the TOE (see ACM_CAP), there must be evidence at ACM_SCP which specific hardware-software pair is used for the TOE.

266 The identification and listing of modules of the design in the configuration list seems to be difficult to apply to ICs. Since, within the framework of the design, functional blocks can be taken out of a developer's HDL or CAD library and out of the IC manufacturer's technology library (lists of the technology parameters). At the very least the libraries as a whole that are used, together with the possible parameters that are used, must be clearly identified if individual library components do not have their own identifier.

267 Test information covered by CM shall include all of the parts which are necessary for the generation and testing of the TOE. That includes all test equipment, libraries, and the list of test vectors used during testing as well as the set of tests including test data and results.

2.7.3 Configuration Management automation (ACM_AUT)

2.7.3.1 Objectives

268 Configuration management automation establishes the level of automation used to control the configuration items. The objective is to determine whether changes to the TOE implementation representation are controlled with the support of automated tools, thus making the configuration management system less susceptible to human error or negligence.

2.7.3.2 Input

269 The developer shall provide a CM plan, including information about the CM-tools (from ACM_AUT.1/EAL4) and use of the CM system.

2.7.3.3 Requirements

270 The TOE implementation representation (hardware schematics and layout) is identified in the IC database. Any change to this database is supposed to be controlled with the support of automated tools. These tools shall be used to ensure the integrity and the access control policy of the database source files: authorized personnel only get access to the database.

271 The evaluator looks for evidence that the tools and procedures are in use. This requires a visit to the development site (see section ALC_DVS).

272 Design documents are presented in the form of data files. Tool-supported configuration control can succeed in any case, at the level of the data files, as it is also implemented in software development environments. All of the relevant data files detailing all of the development steps must be included for the purpose of reproducibility.

273 ACM_AUT.1.2C/EAL4 requires that generation be supported by an automated means. For the IC generation process, it is important that tool support bring together the right hardware-software pair, which means that the right mask must be used. This in turn means that it must be ensured that the mask was demonstrably created from the correct software image. Additionally, tool support is usually given, if the generation process makes use of the appropriate design tools (HDL / CAD tools).

2.8 Delivery and operation (Class ADO)

274 The assurance class ADO defines requirements for the measures, procedures, and standards concerned with secure delivery, installation, and operational use of the TOE, ensuring that the security protection offered by the TOE is not compromised during transfer, installation, start-up, and operation.

2.8.1 Delivery (ADO_DEL)

2.8.1.1 Objectives

275 The objective of these requirements is to determine whether the delivery procedures are documented and maintain integrity and the detection of modification or substitution of the TOE when distributing the TOE to the user's site. It includes special procedures or operations required to demonstrate the authenticity of the delivered TOE. Such procedures and measures are the basis for ensuring that the security protection offered by the TOE is not compromised during transfer.

2.8.1.2 Input

276 The developer shall provide delivery procedures of the TOE or parts of it to the user (from EAL2). The procedures shall be described and used.

2.8.1.3 Requirements

277 During the life-cycle of the hardware TOE, delivery procedures from one site to another are examined: this includes intermediate deliveries from one step to another of the development and manufacturing process and the final delivery to the end-user; the following sites are concerned:

- development of dedicated software and hardware (design centre),
- reticles manufacturer (mask manufacturer),
- manufacturing (fabrication site),
- testing (testing site),
- packaging (microassembly and testing).

278 The examination of the delivery process to determine that the delivery procedures are used, is normally done during site inspections. Subcontractor procedures shall be included. Traceability of what has been delivered by whom to whom is verified. A particular attention is placed on "parallel deliveries" such as samples for quality inspection, scrapped samples, screened processes.

279 If applicable, delivery procedures approved by the national certification body should be followed.

280 Note that a fab-key protection mechanisms may be used to protect the delivery of the chip from the chip manufacturer to the Card Manufacturer/personalization centre.

281 Depending on the scope of the TOE, bringing together the right hardware-software pair can partly be an aspect of the TOE configuration management or of the delivery procedures (see chapter 3.7, ACM). In the case of a modular evaluation approach, the TOE may consist of hardware and Test-ROM software only. Integrating a specific operating system mask file into the chip production process is an aspect covered by ADO_DEL. In this case, the procedures have to be agreed upon between the operating system software developer and the chip manufacturer. Initialization is treated accordingly (initialization data coming from the card manufacturer).

282 For ICs the security state of the chip (test mode, user mode) during delivery is of importance. Thus, functionality for the deactivation of test hardware or for the transition from test mode or installation mode to user mode/operational mode, should be considered and described since this is important in the context of vulnerability analysis and the authenticity of the delivered TOE (see also ADO_IGS).

283 In the context of TOE delivery, tests of security-relevant functionalities are necessary. The results of these tests must be documented. This must be a part of the delivery procedure.

2.8.2 Installation, generation and start-up (ADO_IGS)

2.8.2.1 Objectives

284 The objective of these requirements is to determine whether the procedures and steps for the secure installation, generation, and start-up of the TOE have been documented and result in a secure configuration.

285 This requires that the copy of the TOE be configured and activated by the administrator to exhibit the same protection properties as the master copy of the TOE. The installation, generation, and start-up procedures provide confidence that the administrator will be aware of the TOE configuration parameters and how they can affect the TSF.

2.8.2.2 Input

286 The developer shall provide procedures necessary for the secure installation, generation and start-up of the TOE. The procedures shall be described and used.

2.8.2.3 Requirements

287 For a hardware TOE, procedures for generation of the TOE are examined through life-cycle and configuration management activities.

288 The documentation provided for secure installation and start-up may correspond to documentation of the reset mechanisms (e.g. chip internal power-on-reset procedure, answer to reset at the external interface lines). A secure initial condition for the IC should be determined.

289 Any configuration or administration during life-cycle or operation is to be documented. Configurations on the TOE could be certain modes, such as the test and operational modes, or a choice of certain options through programming of the IC.

290 If not covered under different aspects (see ADO_DEL and ATE_FUN), functionality for the deactivation of test hardware or for the transition from test mode or installation mode to user mode/operational mode should be considered and described here.

291 The procedure of injecting a fab-key should be described.

292 If the TOE definition comprises any initialisation phase, these procedures shall be documented (e.g.. downloading of any embedded code during embedding or personalization). A visit to the embedding and personalisation sites could be necessary; when these phases are included in the limits of the evaluation, corresponding site inspections are mandatory. Otherwise, when this initialisation phase is not included in

the limits of the TOE, these procedures shall be evaluated as part of the administrator guidance if applicable for the TOE as defined.

293 There is no requirement for hardware diagnostic procedures at any CC EAL. Such requirements may nonetheless be expressed as Security *Functional* Requirements using the FPT_AMT.1 or FPT_TST.1 functional components.

294 Trusted recovery is not a requirement of EAL5. This may also be expressed as a Security *Functional* Requirement using FPT_RCV components.

295 If, in the context of the assumed threats, there is a possibility that attacks on a TOE could be successful even if it is finally out of operation, then the termination phase of the use of the TOE should be included in the evaluation (e.g. withdrawing and destruction of a chip or the object reuse of memory areas on the chip).

2.9 Guidance Documents (Class AGD)

296 The assurance class AGD defines requirements directed at the understandability, coverage and completeness of the operational documentation provided by the developer. This documentation, which provides two categories of information (for users and administrators) is an important factor in the secure operation of the TOE.

2.9.1 Administrator guidance (AGD_ADM)

2.9.1.1 Objectives

297 The objective of these requirements is to determine whether administration documentation describes how the administrators administer the TOE in a secure manner and how to make effective use of the TSF privileges and protection functions. The requirements help ensure that the environmental constraints can be understood by administrators and operators of the TOE.

2.9.1.2 Input

298 The developer shall provide administrator documentation. The datasheet of the IC could be considered to support these requirements.

299 Depending on the definition of administrator, administrator and user guidance could be merged in the same document.

2.9.1.3 Requirements

300 During the TOE life-cycle, the administrator role shall be clearly identified. The PP shall be refined in that direction.

301 In most cases, the IC leaves the manufacturer with all of the adjustments, which concern hardware security, and without a change to this adjustment being necessary or even possible.

302 If applicable to the TOE, personalization of the IC can be considered as system administration. In this respect, administration guidance shall document any procedures and mechanisms that guarantee the integrity of the TOE during embedding and personalisation. All of the controllable security parameters and events relevant to

security need to be documented. This may include how to apply the fab-key authorization procedure by the administrator.

2.9.2 User guidance (AGD_USR)

2.9.2.1 Objectives

303 The objective of these requirements is to determine whether user guidance describes the security functions and interfaces provided by the TSF and whether this guidance provides instructions and guidelines for the secure use of the TOE.

304 Requirements for user guidance help ensure that users are able to operate the TOE in a secure manner (e.g. the usage constraints assumed by the PP or ST must be clearly explained and illustrated). User guidance is the primary vehicle available to the developer for providing TOE users with the necessary background and specific information on how to correctly use the TOE's protection functions. User guidance must do two things: firstly, it needs to explain what the user-visible security functions do and how they are to be used, so that users are able to consistently and effectively protect their information; secondly, it needs to explain the user's role in maintaining TOE security.

2.9.2.2 Input

305 The developer shall provide user documentation. The datasheet of the IC could be considered to support these requirements.

306 Depending on the definition of the users, administrator and user guidance could be contained in the same document.

2.9.2.3 Requirements

307 During the TOE life-cycle, the user role shall be clearly identified. The PP shall be refined in that direction.

308 The user guidance shall specifically address the secure usage of the IC security features to cater for software (e.g. the smart card operating system or a smart card application) developers. All recommendations for the use by the software of security mechanisms shall be explicitly stated.

309 The hardware designer as the user of the TOE may either implement the IC in hardware circuitry (e.g. in a terminal if the TOE is a secure application module for the terminal) or uses the IC for an application. He needs details of the IC and its security properties with the intention of the Security Target, in order to be able to transfer the security policy correctly.

310 An end-user of the IC has no direct contact with the IC hardware, but rather uses the chip within the framework of an application. Therefore no documentation of the security properties of the hardware is necessary for this end-user.

311 The information for an IC user should usually be found in a technical IC data sheet (e.g. description of the functionality, pin out, timing diagrams, as well as programming guidelines). For a secure use of the TOE it is necessary, in accordance with CC requirements, to document details of the security properties, special applications advice in the sense of the security policy and advice for the use of SEFs.

312 External interface specifications of the IC to be provided for the user documentation should be taken from the CC functional specification or vice versa, whichever is specified first.

2.10 Life cycle support (Class ALC)

313 Assurance class ALC defines requirements for assurance through the adoption of a well defined life-cycle model for all the steps of the TOE development, including flaw remediation procedures and policies, correct use of tools and techniques and the security measures used to protect the development environment.

314 In general, life cycle support for the TOE is also supported by Configuration Management (class ACM) and delivery and operation aspects (class ADO), see above.

2.10.1 Development security (ALC_DVS)

2.10.1.1 Objectives

315 The objective of these requirements is to determine whether the development and manufacturing environment procedures are adequate to provide the confidentiality and integrity of the TOE design, implementation and production that is necessary to ensure that secure operation of the TOE is not compromised.

2.10.1.2 Input

323 The developer shall provide development security documentation.

2.10.1.3 Requirements

316 During the life-cycle of the hardware TOE, development and manufacturing security procedures are examined. All relevant development and production sites of the TOE must be taken into consideration so that the security requirements are valid for all life cycle phases until the final delivery of the TOE within the scope of the evaluation. This is of particular importance because the requirements on the technology are finally only realized during the production of the ICs, and tests within the framework of the production come into use, too (e.g. wafer tests).

317 The examination includes all the steps of the development and manufacturing process; the following sites are concerned:

- development of dedicated software and hardware (design centre),
- site for creating the image of the application software (if applicable),
- reticles manufacturer (mask manufacturer),
- manufacturing (fab site),
- testing (test site),
- packaging (microassembly and testing) depending on the scope of the TOE.

318 In specific cases there may be a separate design centre for certain cells that are not TOE specific (e.g. standard CPU cell, standard memory cell). For the requirement

ALC_DVS.1, there may be enough evidence that these pre-defined cells (not security enforcing but possibly security supporting) are functionally correct and integer if appropriate delivery procedures, tests and confidentiality agreements are in place. In this case this design centre would not have to be considered under ALC_DVS. The site of the mask manufacturer can be treated accordingly.

319 All the security operational procedures are being checked. The examination is normally done during site inspections. Subcontractor procedures are also checked. It should be noted that evaluator access to manufacturing site, specialist personnel and tools would be required to support these evaluation activities.

320 The Procedures shall include the following types:

- physical (site security: access controls),
- procedural (granting of access to the development tools, revocation of access, transfer of protected material, admitting and escorting visitors, development security policy ...),
- personnel (screening process for new development staff ...),
- IT security measures (Identification and authentication, access control, archiving, audit, networking, firewalls ...).

321 Further sensitive areas within the sites mentioned above might be:

- process control (integration of the circuits onto silicon)
- product engineering (fault analysis with regard to the process)
- quality control for security functionality
- storage / delivery

322 The TOE is present in the various stages of development and production in various different physical shapes. The integrity of the layout masks is of particular significance. Secure delivery will support this aspect (see ADO_DEL).

323 Physical, procedural, personnel and other measures necessary for the realization of the TOE's security properties, as given in the Security Target, will be transposed in the development and production and will have an effect on the security in the operational phase of the TOE. These measures are also to be documented and examined. Measures in the test phase, as well as the assembly phases if applicable, and measures for the management of the manufacturing process are particularly important.

324 In comparison, compilation of a software TOE takes place with fixed options uniquely in the development environment (prototype and master copy). The series production of the software is simply a process of copying, in which the integrity aspects of the copy with respect to the master copy play a role.

325 With respect to IC TOEs, drawings and associated data files will be created within the framework of the development. The IC production of the prototype as well as the series is essentially more complex than a copying process in the case of software and is

variable through a multitude of process parameters, which are potentially manipulable by personnel.

326 The test phase during IC production is of particular importance, since in this phase an IC is already completely physically available, but, for example, internal IC structures are, however, adjustable or compromisable via a test mode, which is still activated.

327 Measures, which are taken, in order to mark (ink) the faulty dice on the wafer and to sort out faulty TOEs (final parts), including which criteria to select, can be of importance. Measures for the destruction of defective parts should then be described.

2.10.2 Flaw remediation (ALC_FLR)

2.10.2.1 Objectives

328 Flaw remediation ensures that flaws discovered by TOE consumers will be tracked and corrected while the TOE is supported by the developer. While future compliance with the flaw remediation requirements cannot be determined when a TOE is evaluated, it is possible to evaluate the procedures and policies that a developer has in place to track and repair flaws, and to distribute the repairs to consumers.

2.10.2.2 Input

329 The developer shall document the flaw remediation procedures.

2.10.2.3 Requirements

330 This CC family is not mandatory for the predefined EAL packages. Nevertheless, it is possible to select one of the ALC_FLR assurance components within the ST by EAL augmentation.

331 Aspects of flaw remediation might be combined with the use of an evaluation maintenance programme supported by the Assurance Maintenance class (AMA) from the CC.

2.10.3 Life cycle definition (ALC_LCD)

2.10.3.1 Objectives

332 The objective of these requirements is to determine whether the developer has used a documented model of the TOE life-cycle for development and maintenance.

333 Life cycle definition establishes that the engineering practices used by a developer to produce the TOE include the considerations and activities identified in the development process and operational support requirements.

334 Confidence in the correspondence between the requirements and the TOE is greater when security analysis and the production of evidence are done on a regular basis as an integral part of the development process and operational support activities. It is not the intent of this component to dictate any specific development process.

2.10.3.2 Input

335 The developer shall provide a life-cycle definition (from EAL4).

2.10.3.3 Requirements

336 The description of the model should include information on the procedures, tools and techniques used by the developer for development and maintenance of the TOE.

337 An example of the life-cycle model is detailed in the Integrated Circuit Hardware Evaluation Methodology [PP-9806]. This model shall be refined by the developer.

338 A basic description is required at ALC_LCD.1 / EAL4, whilst at ALC_LCD.2 / EAL5 the life-cycle model used must additionally be a ***standardized*** model, i.e. one that has been approved by academic experts or standards' bodies.

2.10.4 Tools and techniques (ALC_TAT)

2.10.4.1 Objectives

339 The objective of these requirements is to determine whether the developer has used well-defined tools to develop, analyse and implement the TOE (e.g. programming languages or computer-aided design (CAD) systems) that yield consistent and predictable results.

340 It includes requirements concerning the development tools and implementation dependent options of those tools.

2.10.4.2 Input

341 The developer shall provide development tools' documentation being used for the TOE (from EAL4).

2.10.4.3 Requirements

342 The evaluation aspect for tools and techniques is applicable to software as well as hardware TOEs.

343 When considering tools and techniques for software TOEs, it is a question of getting evidence whether the development tools which have been used are unambiguously and well defined and documented, and whether all options of the tools have been documented. From ALC_TAT.2/EAL5 implementation standards have to be applied. The objective here, apart from a higher assurance into the correct implementation of the TOE, is also to ensure repeatability of the construction of the TOE. The required confirmation of the evaluator that the implementation standards have been applied may require visiting all relevant sites. Therefore, it should be noted that evaluator access to manufacturing site, specialist personnel and tools would be required to support these evaluation activities.

344 For the dedicated software of the IC, this corresponds to the software development tools.

345 The use of the development tools shall be documented. This includes in particular the description of the compiling chain for the software source code (if applicable) and the synthesis chain for HDL. All options shall be documented and parameters shall be clearly identified. The evaluator shall verify this tools' chain, usually during site inspection.

346 In order to achieve the objective of ALC_TAT for a hardware TOE, it is necessary to document and test the hardware description languages (HDL), representation elements (graphical logic elements) and tools (e.g. HDL-compiler, simulation tools, CAD tools) used in the hardware. Additionally supporting libraries shall be considered. A clear and precise definition of all elements and the options used for the TOE is important.

347 In the case of software, various compilers can create different object codes even with the same functionality of the TOE (i.e. with the same logical design). Functionality is defined by the processor commands and the compiler options that are used.

348 In the event that microchip ICs have different masks, differing physical implementations can arise as a result of different technologies, even though functionality may be the same (i.e. with the same logical design at the level of the circuit diagram). Functionality is defined finally only by means of the cell structure which has been implemented in the silicon. As a result of this, the technology used for the implementation of the chip has to be specified. At high assurance levels, the parameters of the technology used have to be documented.

349 For the purpose of clarification, the following table shows the development processes of hardware ICs and software:

Software	Hardware
Program text input via the editor for the creation of the source file. Syntax and semantics of the input language will be determined via the compiler.	Creation of the logical plan through graphic input via a CAD tool or text input via a HDL-Editor. Syntax and semantics of the graphics symbols are determined via the CAD tool and the technology. For the modelling and the simulation of the circuit design a HDL will be used.
In order to construct a functional software TOE, several steps are required: Determining the compiler and linker adjustments, in order, for example, to realise certain optimization possibilities. Compiling and linking of the source files to a program which is executable from a processor during its run time (object files as program data files, run time library, program code for a hardware memory). Testing and de-bugging within the framework of the compilation of individual source files as well as the	In order to construct a functioning IC from the design, several steps are needed: The construction of a netlist out of the logical plans and the synthesis of the gate structure as well as the test structure. The simulation of this logic at the gate level and at the level of the layout using timing defaults. The construction of the layout and the masks. The manufacture of the microchip from this masks after several process steps, which are dependent on the semiconductor technology used (e.g.

Software	Hardware
whole TOE.	0.8_ìm CMOS-, BiCMOS- or Bipolar technology).

Table 3 - Development processes of hardware ICs and software

350 A programming language used is based on the features of a compiler, interpreter or assembler, while the logic which has been constructed using an HDL is finally only available after the conclusion of the technological process. Therefore, this assurance aspect is to be understood in the sense of ”Tools, Techniques and Technology”.

2.11 Vulnerability assessment (Class AVA)

351 The assurance class AVA defines requirements directed at the identification of exploitable vulnerabilities. Specifically, it addresses those vulnerabilities introduced in the construction, operation, misuse, or incorrect configuration of the TOE.

2.11.1 Covert channel analysis (AVA_CCA)

2.11.1.1 Objectives

352 Covert channel analysis is directed towards the discovery and analysis of unintended communications channels that can be exploited to violate the intended TSP.

2.11.1.2 Input

353 From EAL5 the developer shall conduct a search for covert channels and provide covert channel analysis documentation.

2.11.1.3 Requirements

354 A covert channel analysis is explicitly required by CC, only if the TOE implements an information flow control policy. In practice, this will in turn depend on whether the ST specifies such a policy through the use of functional components from the FDP_IFC and FDP_IFF families.

355 Therefore, covert channel analysis may not be required for an IC itself (but inference channel analysis will almost certainly be required). It may be applicable in a composite TOE evaluation (IC hardware plus operating system and application software and IC dedicated software) where the analysis should discuss the interrelationships between the software and IC, to demonstrate that they are mutually supportive in helping to meet the Security Target for the composite TOE. This will not only involve discussion of dependencies of the IC on the software, but also the software dependencies on the hardware, including tamper-resistance aspects.

356 As an example of software-hardware interdependency, the following example is considered. The software may implement a mechanism enforcing a limit on the number of times a key can be used in order to prevent DPA being fully exploited. However, it will be necessary to determine whether the rate of leakage by the IC is sufficiently low so that the number of samples available from the key is insufficient to reveal the key.

357 Nonetheless, covert channels and inference channels may need to be considered. e.g. covert channels may arise from dynamic interweaving in the timing behaviour of individual security functions or mechanisms.

358 Furthermore, in the case of an IC, covert channel analysis would involve analysis of physical channels using techniques such as Timing Analysis, Simple Power Analysis, and Differential Power Analysis.

359 Design information about physical connections between physical components in the form of signal paths and circuits or in form of the layout (i.e. that information on the technical and technological implementation that needs to have some influence in the analysis) is important to search for covert channels.

360 If the CC ST contains no functional components from the FDP_IFC or FDP_IFF families, it would be appropriate to address issues such as leakage attacks (which might be thought of as covert channel exploitation) under the AVA_VLA heading instead of AVA_CCA.

2.11.2 Misuse (AVA_MSU)

2.11.2.1 Objectives

361 The objective of these requirements is to determine whether misleading, unreasonable and conflicting guidance is absent from the guidance, whether secure procedures for all modes of operation have been addressed, and whether use of the guidance will facilitate detection of insecure TOE states.

2.11.2.2 Input

362 Guidance documentation shall be used for the analysis (from EAL3). From AVA_MSU.2/EAL4 the developer shall document an analysis of the guidance documentation.

2.11.2.3 Requirements

363 User and administrator guidance as well as installation, generation and start-up procedures if applicable shall be considered for this activity.

364 All modes of operation of the TOE shall be considered, their consequences and implication for maintaining secure operation. These set of requirements focus on the effectiveness point of view of the operational guidance. The guidance documents should give an accurate, consistent and complete description of the operational modes of the TOE, and how insecure states may be detected.

365 If, in the framework of the configuration (see ADO_IGS), an administration of hardware security functions or an initialization of the HW is necessary, this should be analysed with regard to preserving the security properties of the TOE, the secrets under protection and with regard to the ease of use for an end-user. Examples are the injection of a fab-key, special external cabling or certain programming, or making certain operational modes such as the initialization and test modes explicitly ineffective by special measures.

366 If there are types of operations among peripheral conditions, like temperature, voltage, frequency etc., which the end-user can possibly influence during operation of the TOE, then the effects on security should be analysed.

367 Dependencies between hardware and software should be addressed as if they are important for the software developer to use hardware security functions in the intended manner.

368 Misuse analysis might also be related mainly to the smartcard manufacturing and production environments, rather than the end-user operational environment, especially in respect of use of personalization equipment.

369 The analysis will reflect any applicable aspects of platform protection, which may involve detection of insecure states. Actions taken on detection of corruption should also be documented (e.g. render a smart card IC unusable or correct failure). Other aspects of relevance may include the operation of on-chip test software.

2.11.3 Strength of TOE security functions (AVA_SOF)

2.11.3.1 Objectives

370 The objective of these requirements is to determine whether Strength of Function (SOF) claims are made in the ST and whether the developer's SOF claims are supported by an analysis that is correct (i.e whether such functions meet or exceed the claim).

371 According to CC the analysis addresses TOE security functions that are realised by a probabilistic or permutational mechanism (e.g. a password or hash function, random number generators) for which some quantitative or statistical analysis can be performed.

372 Even if such functions cannot be bypassed, deactivated, or corrupted, it may still be possible to defeat them by direct attack.

373 As the selection of a certain AVA_VLA-assurance component defines the baseline for the protection of the TOE in terms of attack potential against which the vulnerability analysis of the TOE will be judged, the selected SOF-level can assume a higher attack potential for certain mechanisms (see [CEM], annex A.8).

2.11.3.2 Input

374 The developer shall perform a strength of TOE security function analysis (from EAL2).

2.11.3.3 Requirements

375 SOF analyses put the attack scenarios intended by the ST in concrete terms and examines how the probabilistic or permutational mechanisms specified for the TOE security functions withstand in the face of direct attacks. Therefore, all probabilistic or permutational mechanisms have to be specified by the underlying algorithms, properties and principles at the appropriate level of granularity within the design documentation.

376 Specific mechanisms, which can be attacked typically by using statistical methods and without physical modification of the TOE (e.g. mechanisms countering DPA), can be considered under AVA_SOF, too.

377 SOF analysis can be a question of the following types of direct attack:

- attacks without physical modification of the TOE, they are similar to traditional direct attacks on software but possibly involving physical means
- attacks on the mechanisms implementing technical and technological properties of the TOE without physical modification of the TOE

378 The analysis has to show that each such mechanism satisfies the claimed minimum strength stated in the security target.

379 If a specific direct attack on a TOE security function covered under SOF requires physical modification of the IC as a pre-requisite, the whole attack path has to be considered for the decision about the exploitability using the calculation tables in [AP-SC].

380 In the assessment of the SOF, the following aspects, among others, could have an influence, depending on the security objective:

- active hardware security (e.g. active sensors, I&A etc.)
- hardware security through technical and technological properties (e.g. fixed masks, bus circuits, casings, passivation, E^2PROM cycles etc.)
- supporting mechanisms as firmware/software modules (e.g. check sums)

381 Typically, the mechanisms of concern are susceptible to direct attack if an attacker can defeat it by determining (e.g. through exhaustive attack) the value of a 'secret' or the asset to be protected.

382 [AP-SC] gives guidance for the SOF calculation.

383 CEM requires that any assertion or assumption supporting the analysis be valid be determined. Assumptions should reflect the worst case, unless worst case is invalidated by the ST. Therefore, justification shall be provided regarding how the guidance is used in a certain manner (e.g. why a certain level of expertise or a certain equipment for an attack is applicable and no less).

384 With respect to attacks which physically modify the internal technical structures of the TOE, it is a question of an indirect attack which needs to be examined in the context of a vulnerability analysis, since SFs may be bypassed and therefore may loose their effectiveness.

385 It should be noted that the CC uses the term *attack potential* when defining the SOF levels (low-moderate-high attack potential). However, the three SoF levels defined in CC (SOF-basic, SOF-medium and SOF-high) are directly related to the attack potential.

386 The CC optionally allows specific SOF metrics to be claimed in the Security Target for individual mechanisms, which the Strength of Function Analysis must show are upheld.

387 If the scheme rules allow the rating of cryptographic functions, SOF analysis can be applied accordingly. This might be the case for an encryption algorithm implemented in a hardware crypto engine. In case of a composite TOE evaluation it may be applicable for encryption functions to be implemented partly in hardware and software.

2.11.4 Vulnerability analysis (AVA_VLA)

2.11.4.1 Objectives

388 Vulnerability analysis consists of the identification of flaws potentially introduced in the different refinement steps of the development. It results in the definition of penetration tests through the collection of the necessary information concerning: (1) the completeness of the TSF (does the TSF counter all the postulated threats?) and (2) the dependencies between all security functions. These potential vulnerabilities are assessed through penetration testing to determine whether they could, in practice, be exploitable to compromise the security of the TOE.

2.11.4.2 Input

389 The developer shall perform and document a vulnerability analysis of the TOE (from EAL2). This document shall describe known vulnerabilities and assess their impact.

2.11.4.3 Requirements

390 Vulnerability Analysis in the CC comprises construction and operational vulnerabilities of the TOE. Any ways in which the security features may be deactivated, bypassed or corrupted should be documented by the developer. This analysis must provide arguments as to why the vulnerability cannot be exploited within the TOE's environment.

391 One key aspect of the CC vulnerability analysis requirements is the notion of resistance to attack posed by attackers who have a particular ***attack potential*** (low-moderate-high attack potential) - this comes into consideration at AVA_VLA.2 and above.

392 The scope of the search and the analysis for vulnerabilities which the developer must perform depends on the selected AVA_VLA component:

- obvious vulnerabilities at AVA_VLA.1
- resistance to low attack potential at AVA_VLA.2
- resistance to moderate attack potential at AVA_VLA.3
- resistance to high attack potential at AVA_VLA.4.

393 Evaluators' vulnerability analysis is based on the same level of attack potential.

394 Hardware TOEs can be subject to vulnerabilities which can be exploited by physical tampering of the TOE. Such tampering could circumvent the effectiveness of security enforcing functions. This aspect must be considered during vulnerability assessment and penetration testing.

395 [AP-SC] gives guidance for the rating of the vulnerabilities (low, moderately, highly). Rating provides a measure of the weakest path required to determine IC secrets or tamper with the IC. In practice, this is likely to be the sum of various work functions (e.g. remove protective barrier, determine IC layout, decrypt data or extract EEPROM contents).

396 When using the rating tables from [AP-SC] or [CEM], annex A.8 justification shall be provided why the assertions or assumptions supporting the analysis are valid (e.g. why

a certain level of expertise or a certain equipment for an attack is applicable and no less).

397 Vulnerabilities can be introduced in both the construction of the mechanism itself, as well as through the production of technical and technological measures intended to counter threats. The effectiveness of the functionality depends on the technology used in the implementation phase. This must be taken into consideration in the vulnerability analysis.

398 The process of aging the IC should be taken into consideration during vulnerability analysis. So, for example, the vulnerabilities of an IC TOE can lie in the semiconductor technology. E^2PROM cells only withstand a restricted number of program cycles. The limitation of the number of possible delete and write cycles of a cell is an inherent vulnerability, which an attacker could possibly exploit. This kind of technologically based vulnerability analysis is new in contrast with software TOEs, and requires a vulnerability analysis relating to the technology and its implementation.

399 Nevertheless, vulnerabilities can possibly be exploited by means of the physical tampering of the IC or through influencing the timing of signals to the external interfaces.

400 With respect to attacks which physically modify the internal technical structures of the TOE, it is a question of an indirect attack which needs to be examined in the context of a vulnerability analysis, since SEFs may be bypassed and therefore may loose their effectiveness. Additionally, the binding of distinct components realized in mechanisms has to be taken into consideration.

401 For example, attacks via micro-probes on IC internal signals (white box attack) could possibly be successful after a physical modification of the TOE. Protection structures in the IC, which simply make an attack more difficult, could, however, be fundamentally overcome (e.g. the etching of individual layers, the deactivation of sensors though the masquerade of the logical sensor signal). The exploitability of such indirect attacks must be viewed as a vulnerability. Specialist tools and expertise are necessary for the penetration of a HW-chip.

402 Additional vulnerabilities can be of the following types: non-documented functionalities or hardware areas on the chip, covert/inference channels, radiation, power consumption, timing. Hardware implementation documentation, as well as the HDL bases, should be used for these examinations.

403 The operational vulnerabilities are also to be considered in the context of the use of the chip by an operating system or an application developer. For instance, the security measures which should be taken in the application development and which influence the operation of the IC could possibly be exploited by an attacker (e.g. demands on external cabling, external technical parameters, or confidentiality measures).

404 If the examination of hardware sensors signals is not done by the TOE itself, but needs to be done by the overlying operating system, this can cause an operational vulnerability.

405 Also the possibilities of deliberately provoking faulty conditions or technical defects in the operation of the IC in order to make them exploitable for an attack can be considered under operational vulnerabilities.

406 Note that, exploitation of covert channels is a special form of indirect attack, which may require a separate analysis under CC (see AVA_CCA above).

407 The approach of the CC in the evaluation of vulnerabilities (e.g. for tamper resistance) is that the attack potential considered for a TOE evaluation is pre-defined by selection of a certain AVA_VLA assurance component and not by selection of a SOF-level. Therefore, for IC or smartcard evaluations, if the target EAL is less than EAL6 it may be appropriate to augment the assurance requirement with a higher AVA_VLA component.

408 If resistance to attackers with a (i) moderate / (ii) high attack potential is required and the assurance package EAL4 is selected in the ST, EAL4 must be augmented by (i) AVA_VLA.3 / (ii) AVA_VLA.4 within the ST. If the assurance package EAL5 is used, this can be augmented by the AVA_VLA.4 assurance component to get resistance to attackers with high attack potential. In both cases no additional dependency to other assurance components has to be fulfilled.

409 Binding aspects should be considered during vulnerability analysis, owing to vulnerabilities, which may arise from problems in binding if the TOE's security functions are not mutually supportive or do not provide an integrated and effective whole. In particular, it addresses the issue of indirect attack against the IC (e.g.

- physical connections between physical components in the form of signal paths and circuits
- physical connections between physical components because of the layout (i.e. that information on the technical and technological implementation needs to have some influence in the analysis)
- dynamic interweaving in the timing behaviour of individual security functions or mechanisms
- influence on binding through the setting of external signals on the microchip.

410 See also the discussion of covert channel analysis AVA_CCA above for composite TOE evaluations. Binding of hardware- and firmware-functionalities is to be taken into consideration, depending on the definition of the TOE boundary and the separation of TSP enforcing and non-TSP enforcing parts of the TOE.

411 Further guidance for penetration testing and vulnerability analysis is given in [AP-SC]. Other activities will be performed according to standard CC practice as documented in the CC, CEM, and other Scheme documentation.

412 It is assumed that evaluators will develop an understanding of security for the target of evaluation and gain an appreciation of its liability to vulnerability from a review of the design and implementation documentation. In the case of smartcards, however, out of operational envelope vulnerabilities are unlikely to be evident in design. Hence, it is important that tamper-resistance evaluators have current working knowledge of tamper-resistant attack techniques, tools and silicon technology.

3 Glossary

BiCMOS	Bipolar Complementary Metal Oxide Semiconductor, specific semiconductor technology
CAD	Computer Aided Design
CMOS	Complementary Metal Oxide Semiconductor, specific semiconductor technology
Die(Pl.:Dice)	individual IC on a wafer
EPROM	Erasable Programmable Read Only Memory
E^2PROM	Electrically Erasable Programmable Read Only Memory
Fusing	the calculated melting of a contact on an IC
HDL	Hardware Description Language
HW	Hardware
IC	Integrated Circuit, integrated electronic circuits in a microchip
Passivation	a protection layer on top of the metallisation layer in a microchip
Pin	external contact of a packaged microchip
SW	Software
TOE	Target of Evaluation
Wafer	silicon slice for chip production

4 References

[CC] Common Criteria for Information Technology Security Evaluation, Version 2.3, June 2005

[CEM] Common Methodology for Information Security Evaluation (CEM), Version 2.3, June 2005

[AP-SC] Application of Attack Potential to Smartcards, Version 2.1, CCDB-2006-04-002, April 2006

[PP-SCSUG] Smart Card Protection Profile, Smart Card Security Users Group Draft Version 1.0, 1 November 1999

[PP-9806] Smartcard Integrated Circuit Protection Profile PP/9806, Version 2.0, September 1998

[PP-9810] Smartcard Embedded Software, PP/9810, Version 1.2

[PP-9911] Smartcard Integrated Circuit with Embedded Software PP/9911, Version 2.0