Joint Interpretation Library

Guidance for smartcard evaluation

Version 3.0
April 2024

This page is intentionally left blank

Table of content

1. References

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

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

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

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

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

[CEM] CCMB-2022-11-006: Common Methodology for Information Technology Security Evaluation, CEM:2022, Evaluation Methodology, Revision 1, November 2022

[APSC] Supporting Document (Mandatory Technical Document): Application of Attack Potential to Smartcards and Similar Devices, latest approved version

[JCPE] Supporting Document (Mandatory Technical Document): Composite product evaluation and certification, latest approved version

[ETRC] Supporting Document Guidance: ETR template for composite evaluation of Smart Cards and similar devices, latest approved version

[SARCH] Supporting Document (Mandatory Technical): Security Architecture Requirements (ADV_ARC) for smartcard and similar devices, latest approved version

[SCER] CCDB-2007-11-001 Site Certification. Supporting Document (Guidance), latest approved version

[MIRSE] Supporting Document (Mandatory Technical): Minimum ITSEF Requirements for Security Evaluations of Smart cards and similar devices latest approved version.

2. Objective

1 This document defines smartcard evaluation and certification terminology and describes appropriate advice.

2 The main intended audience of this document are evaluation sponsors and manufacturers, but it is also related to evaluators, certifiers and end users of this type of products.

3. Smartcard product presentation and definitions

3.1. Glossary

3 The following definitions are used throughout the document. It is important that each term be clearly understood in order that guidance documentation for the evaluation process be put in context.

3.1.1. Integrated Circuit (IC)

4 Electronic component(s) designed to perform processing and/or memory functions (i.e. the hardware component containing the micro-controller and IC dedicated software).

5 A typical IC comprises: a processing unit, security components, I/O ports and volatile and non-volatile memories. It also includes any IC designer/manufacturer proprietary IC dedicated software, required for testing purposes. This IC dedicated software may be either IC embedded software (also known as IC firmware) or security-relevant parts of tests programs outside the IC. The IC may include any IC pre-personalization data.

6 Figure 1 below describes a typical IC and smartcard product hardware architecture:

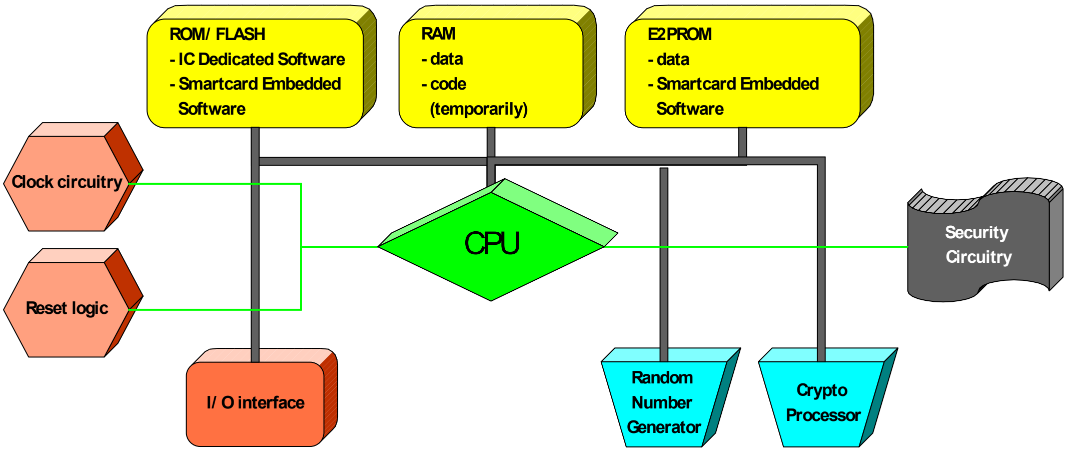


Figure 1 – Typical Smartcard IC

3.1.2. IC Dedicated Software

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

3.1.3. IC Dedicated Test Software

8 That part of the IC Dedicated Software (refer to above) which is used to test the device but which does not provide functionality during Phases 4 to 7. (Phases are described in figure 4)

3.1.4. IC Dedicated Support Software

9 That part of the IC Dedicated Software (refer to above) which provides functions in

Phases 4 to 7. The usage of parts of the IC Dedicated Software might be restricted to certain phases.

3.1.5. Identification Data

10 Any data defined by the Integrated Circuit manufacturer and injected into the non-volatile memory by the Integrated Circuit manufacturer (Phase 3). These data are for instance used for traceability.

3.1.6. Basic Software (BS)

11 Smartcard embedded software in charge of generic functions of the Smartcard IC, such as an operating system, general routines and interpreters.

3.1.7. Application Software (AS)

12 Smartcard embedded software (may be in ROM or loaded onto a platform in EEPROM or Flash Memory) This is software dedicated to the applications.

3.1.8. Embedded Software (ES)

13 Software embedded in a smartcard IC but not developed by the IC Designer. This comprises embedded software in charge of generic functions of the Smartcard IC, such as an operating system, general routines and interpreters (Smartcard Basic Software -BS) and embedded software dedicated to applications (Smartcard Application Software - AS). The Smartcard Embedded Software is designed in Phase 1 and embedded into the Smartcard IC in Phase 3 or in later phases of the smartcard product life-cycle.

3.1.9. Smartcard Personalization

14 Final process under the responsibility of the card issuer, through which a smartcard is to be configured, security parameters loaded and secret keys set. At the end of the personalization process, the smartcard is irreversibly set into “user mode”. Hence, it becomes fully operational and can be delivered to the end user.

3.1.10. IC Platform

15 Usually refers to a smartcard component which may undergo an evaluation process, as a complete Target of Evaluation (TOE) in itself, but which is not an end-user product (i.e. a smartcard component without any Application Software loaded).

3.1.11. IC Pre-personalization

16 Process performed at the IC manufacturer site, through which customer data can be loaded onto the IC, prior to the IC being irreversibly set into “issuer mode”.

3.1.12. IC Pre-personalization data

17 Any data supplied by the software developer that is injected into the non-volatile memory by the Integrated Circuits manufacturer (Phase 3). These data are for instance used for traceability and/or to secure shipment between phases.

3.1.13. Smartcard product

18 A product corresponds to a fully operational smartcard, composed of both IC and complete ES, including application software as appropriate.

3.2. Architecture

19 The figures 2 and 3 below describe typical smartcard product architectures:

3.2.1. Closed architecture

20 All applications that are in the smartcard are known at the time of the evaluation.

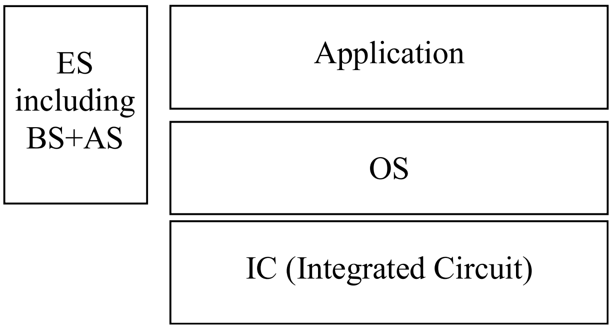


Figure 2 – Typical Smartcard architecture (Closed architecture)

3.2.2. Open architecture

21 New applications could be accepted after the emission of the card.

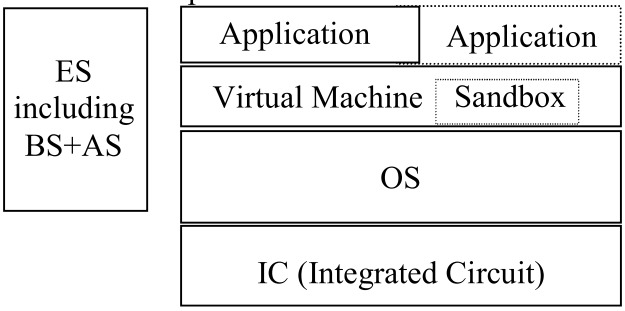
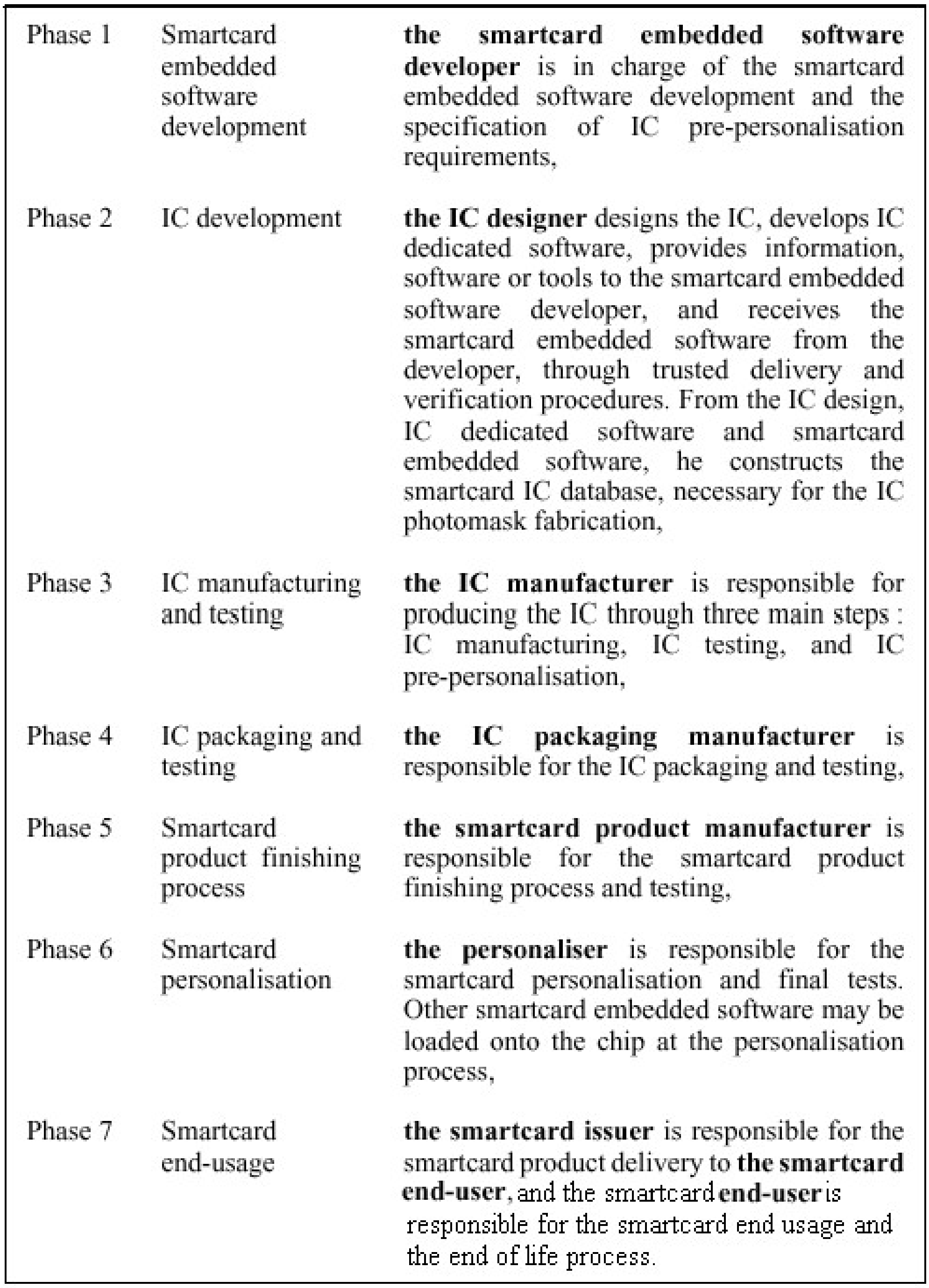


Figure 3 – Typical Smartcard architecture (Open architecture)

22 An application running over the Virtual Machine could be using a “sandbox” or security domain in terms of ADV_ARC. The use of this “security domain” concept in the smartcard field is discussed in the document “JIL – Security Architecture Requirements (ADV_ARC) for smartcard and similar devices” [SARCH].

3.3. Smartcard product life-cycle presentation

23 Figure 4 below describes the smartcard product life-cycle, which can be decomposed into 7 phases where the following authorities are involved:



Phase 1	Smartcard embedded software development	**the smartcard embedded software developer** is in charge of the smartcard embedded software development and the specification of IC pre-personalisation requirements,
Phase 2	IC development	**the IC designer** designs the IC, develops IC dedicated software, provides information, software or tools to the smartcard embedded software developer, and receives the smartcard embedded software from the developer, through trusted delivery and verification procedures. From the IC design, IC dedicated software and smartcard embedded software, he constructs the smartcard IC database, necessary for the IC photomask fabrication,
Phase 3	IC manufacturing and testing	**the IC manufacturer** is responsible for producing the IC through three main steps : IC manufacturing, IC testing, and IC pre-personalisation,
Phase 4	IC packaging and testing	**the IC packaging manufacturer** is responsible for the IC packaging and testing,
Phase 5	Smartcard product finishing process	**the smartcard product manufacturer** is responsible for the smartcard product finishing process and testing,
Phase 6	Smartcard personalisation	**the personaliser** is responsible for the smartcard personalisation and final tests. Other smartcard embedded software may be loaded onto the chip at the personalisation process,
Phase 7	Smartcard end-usage	**the smartcard issuer** is responsible for the smartcard product delivery to **the smartcard end-user**, and the smartcard **end-user** is responsible for the smartcard end usage and the end of life process.

Figure 4 – Smartcard product life-cycle and associated responsibilities

24 Note 1: Sometimes the IC manufacturer delivers modules ready for physical embedding into a plastic card. In this case, they are mostly in charge of Phase 4, i.e. IC packaging manufacturer's duty.

25 Note 2: The generic life cycle described in previous figure 4 can be adapted to consider the opportunity of installing or managing security domains (for both close and open architecture smartcards). For details about the use of security domains in smartcards see the document "JIL – Security Architecture Requirements (ADV_ARC) for smartcard and similar devices" [SARCH].

26 Note 3: In case of having an open architecture e.g. with a Java Card, the life cycle described in figure 4 has to consider to include the development of applications, in addition to the IC and operating system, and how these applications can be loaded and personalized after the platform in order to be used by the end-user. See figure 5 below.

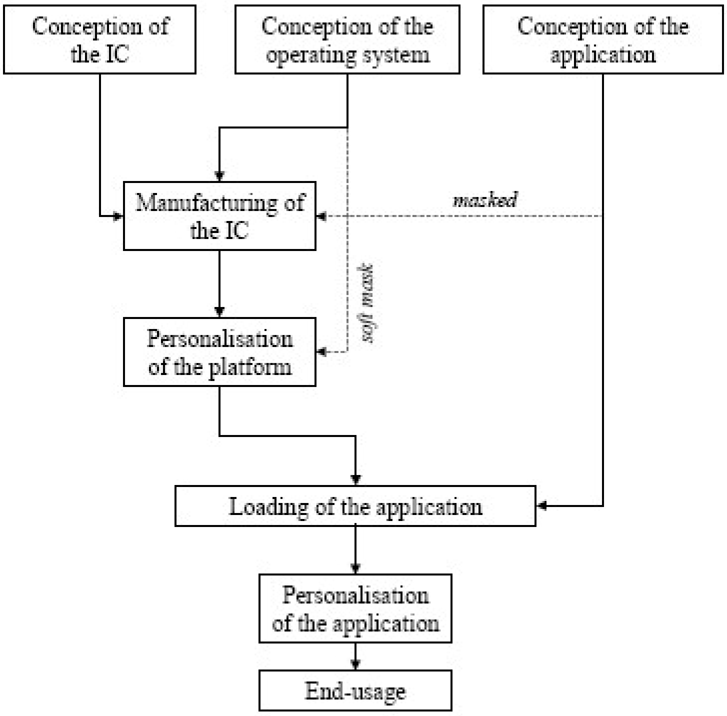

Conception of the IC
Conception of the operating system
Conception of the application
Manufacturing of the IC
masked
soft mask
Personalisation of the platform
Loading of the application
Personalisation of the application
End-usage

Figure 5 – Open platform smartcard life-cycle

4. Contributors roles in product evaluation

4.1. Roles clarification

27 Depending on the exact TOE scope, the following entities described below may be involved in a smartcard evaluation process.

4.1.1. IC Manufacturer

28 Institution (or its agent) responsible for the IC manufacturing, testing, and pre-personalization.

4.1.2. ES Developer or AS Developer

29 Institution (or its agent) responsible for the smartcard Embedded Software or Application Software development and the specification of IC pre-personalization requirements.

4.1.3. Card Manufacturer

30 The customer of the IC Manufacturer who receives the TOE during TOE Delivery. The Card Manufacturer includes all roles after TOE Delivery up to Phase 7. The Card Manufacturer has the following roles: (i) the Smartcard Product Manufacturer (Phase 5); (ii) the Personalizer (Phase 6). If the TOE is delivered after Phase 3 in the form of wafers or sawn wafers (dice) they also assume the role of the IC Packaging Manufacturer (Phase 4). Usually, the Card Manufacturer is also the ES or AS developer.

4.1.4. Card Issuer

31 Customer for a product who is in charge of the issuance of the product to the smartcard holders (end users).

4.1.5. Sponsor (of the evaluation)

32 This is the body responsible for requesting and usually financing an evaluation: candidates might be the developer of the Target of Evaluation, the card issuer or even an independent customer of the product.

4.1.6. Evaluator

33 The evaluation laboratory that performs the evaluation work under a national scheme.

4.1.7. Certification body or Evaluation Authority

34 An independent overseer that licenses national evaluation laboratories and issues the certificate based on the work of such laboratories.

4.1.8. End User

35 The card user acting as operator of the smartcard TOE.

4.2. Steps to be performed in order to get ready for an evaluation

36 The following steps need to be performed in order to prepare for an evaluation:

37 In order to provide the evaluator with the required deliverables, the sponsor and developer need to make sure:

- they have the appropriate skills and manpower,
- an adequate development methodology is used,
- an appropriate development environment has been set up.

38 Alternatively, the developer could obtain training and/or assistance in the field of evaluation criteria. In this respect, evaluation consultancy might be elicited to assess a TOE's ability to meet its evaluation target (e.g. determine any evaluation deliverable shortfall or evaluation constraints).

39 The sponsor (possibly assisted by developer) has to make it clear that they know precisely what they want to be evaluated (IC, ES, platform, application software or any combination thereof).

40 The sponsor (possibly assisted by the developer) might choose to invoke an existing Protection Profile.

41 The sponsor (normally assisted by the developer) has to make available a Security Target which is precise and unambiguous, in order to ensure all relevant parties know exactly what is to be evaluated and against which requirements. Each party must approve the Security Target, thus reducing the risk of evaluation slippage.

42 The sponsor needs to select an evaluation laboratory responsible for carrying out the evaluation. This is normally accomplished through an Invitation To Tender or direct contact with a specific evaluation laboratory. The price and evaluation time scales and any relevant non-disclosure agreements are factors to be negotiated.

43 The sponsor needs to make sure all parties are made aware of the role they play in the evaluation process, particularly regarding expected delivery date and content of evaluation deliverables.

44 Regarding the documentation needed for the evaluation, the objective is to reuse existing developmental documentation as much as possible. The shortfall between developer deliverables and CC requirements must be identified. Where a sponsor is new to evaluation, it is recommended that an evaluation laboratory be commissioned to perform a fit-for-purpose assessment of evaluation deliverables.

45 For an EAL3 or higher evaluation, sponsor and developer need to be aware before the evaluation starts of site security requirements. The ALC_DVS family on Development Security is included in EAL3-EAL7.

46 For a first of kind evaluation, it is recommended that a pre-evaluation review of sponsor deliverables and site security be performed.

47 Figure 6 below presents the above steps to be performed:

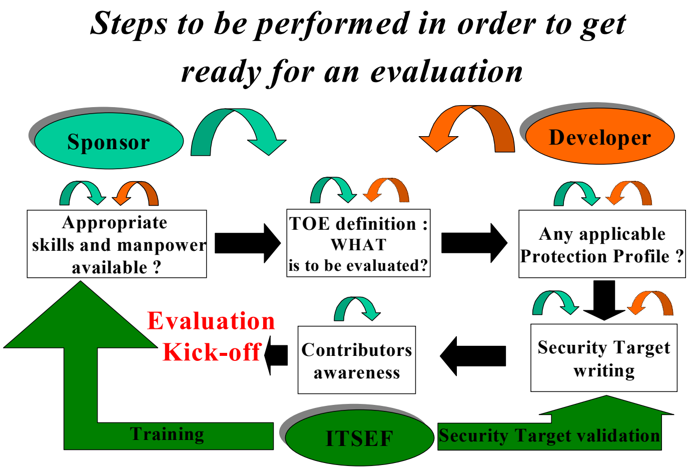

Steps to be performed in order to get ready for an evaluation
Sponsor
Developer
Appropriate skills and manpower available ?
TOE definition : WHAT is to be evaluated?
Any applicable Protection Profile ?
Evaluation Kick-off
Contributors awareness
Security Target writing
Training
ITSEF
Security Target validation

Figure 6 – Steps to be performed to be ready for an evaluation

4.3. Contributors involvement

48 The scope of evaluation concerns either an IC or an IC with embedded software.

49 The following contributions to the evaluation process are expected from each party:

4.3.1. IC manufacturer

50 The IC manufacturer is involved in the evaluation process in case the evaluation scope, includes the IC.

51 He provides the evaluation sponsor with the Security Target for the IC (for approval purposes), if requested to do so and provided it does not compromise IC proprietary content; otherwise, only part of the Security Target may be delivered to the evaluation sponsor (i.e. usually the first chapters of the Security Target without the Rationale).

52 If an IC with ES evaluation is required, the global Security Target must be based on the IC Security Target, which is thus required to be delivered to the global Security Target writer (see concept of ST-lite).

53 He provides the evaluation laboratory with the entire Security Target for the IC (mandatory for an IC evaluation).

54 He provides the evaluation laboratory with every required evaluation deliverable according to the targeted evaluation level and evaluation scope, as defined in the Security Target.

4.3.2. ES developer or AS developer

55 The ES or AS developer is involved in the evaluation process, when the evaluation scope includes ES or AS.

56 He may be requested by the evaluation sponsor to write or assist him write the Security Target.

57 He provides the evaluation laboratory with every required evaluation deliverable

according to the targeted evaluation level and evaluation scope, as defined in the Security Target.

58 He provides IC pre-personalization data.

4.3.3. Card Issuer

59 The card issuer is the customer for a product. They are responsible for the issuance of the product to the smartcard holders; as a minimum, they should be involved in the evaluation process and assumes responsibility for:

- Security Target approval,
- definition of the smartcard personalization data for the product,
- and authorship of the smartcard product guidance documentation.

4.3.4. Sponsor (of the evaluation)

60 The sponsor of the evaluation is involved in the evaluation process is responsible for:

- writing and/or approving the Security Target (he can ask the developer to write the Security Target for him, but they need to approve its contents because it is the baseline for the whole evaluation process),
- mainly ensuring that every required evaluation deliverable be made available to the evaluator.

4.3.5. Evaluator

61 The evaluator analyses the evidence elements they are provided with throughout the evaluation process:

- They perform functional and penetration testing on the TOE.
- They perform a site visit to the development premises.
- They perform a site visit to the production premises (for the evaluation including the IC).
- They write and issue evaluation reports (including the final Evaluation Technical Report – ETR).

4.3.6. Certification body or Evaluation Authority

62 The certification body (CB) acting as evaluation authority is involved in any evaluation process running under its own scheme. It is responsible for:

- approving the evaluation scope as defined in the Security Target before the evaluation process is allowed to start,
- giving advice regarding the evaluation of cryptographic aspects,

- monitoring the evaluation work performed by the evaluation laboratory throughout the evaluation process (evaluation results and evaluation reports approval, attending evaluation meetings etc.),
- and finally, issuing a certificate and a certification report (assuming the evaluation process leads to an overall "Pass" verdict).

4.4. Detailed contributors inputs and evaluator tasks during the evaluation process

4.4.1. General evaluation inputs definition

63 Regarding the documentation needed for the evaluation, it is recommended to provide one "header" document for each task of a CC evaluation. For example, the "header" document could link internal documents with CC tasks. In some cases, more than one document is required like tasks related to the class ALC in families ALC_CMC, ALC_CMS, and ALC_DVS components where one set of documents per site is required.

64 For tutorial purposes, we are going to illustrate the theory using a specific evaluation level. As an instance, the guidance described in this document has been developed according to one common evaluation level used in various operational contexts and experiences i.e. the EAL4+ augmented with ALC_DVS.2 and AVA_VAN.5, in terms of Common Criteria version 3.

4.4.2. EAL4+ evaluation: Contributors inputs and evaluation tasks

Scope IC alone:

65 The only contributor involved in a hardware evaluation besides the evaluators and certification body, is the IC manufacturer.

66 As the IC certification purpose is to be reused in an IC with ES evaluation, some documents need to be produced for composite evaluation purposes (i.e. the ETR_COMP is issued from the full IC ETR, see [ETRC].). This is the case described in the next point of this section.

Scope IC with ES (IC being already certified):

67 The evolution of smartcard technology towards open operating systems, lead to the consideration of a modular approach that takes into account a three basic layers product architecture: the integrated circuit (IC), and the ES consisting of operating system (OS) and application. This modular approach has as objective to allow the most re-usability in composite evaluations.

68 The recommended evaluation strategy is to do a first evaluation with the IC alone, followed by a second evaluation of the OS with the IC or the application and OS with the IC. For example, in the case of a Java Card the optimum approach would be: first evaluate the IC, second evaluate the OS and third evaluate the application. This strategy provides the optimum re-usability and the trust in the complete product security. The second evaluations could be reduced to the activities dedicated to the upper layer with a re-use of the lower layers evaluation results. Other strategies based on evaluate the OS and/or application without the IC would be a significant step for enforcing OS correctness but

will not give a complete view on the product robustness.

69 For more details on the composition of an ES with an IC already certified, see the document “JIL Composite product evaluation for Smartcards and similar devices” [JCPE] and the “ETR for composite evaluation” template [ETRC].

Annex A Theoretical planning for an EAL4+ evaluation

A.1 Foreword

70 This annex gives a theoretical planning for an EAL4+ smartcard evaluation from the Evaluation laboratories point of view. For example, it could be an evaluation conformant to the Smartcard Integrated Circuit with Embedded Software Protection Profile.

71 The evaluation tasks are sequenced as quickly as possible, according to the idealistic assumptions:

- The IC evaluation is not considered in this planning as results are already available through the hardware platform evaluation.
- Parallelism is included:
    - this relies on the hypothesis that an infinite number of evaluators are available for the evaluation with a good knowledge of the product;
    - the developers provide deliveries on time without delay;
- Deliveries iterations are not considered. The hypothesis is that there is no critical issue that stops the progress of the evaluation.

72 On a fact based experience, a **6 months evaluation** are **achievable** under the following conditions:

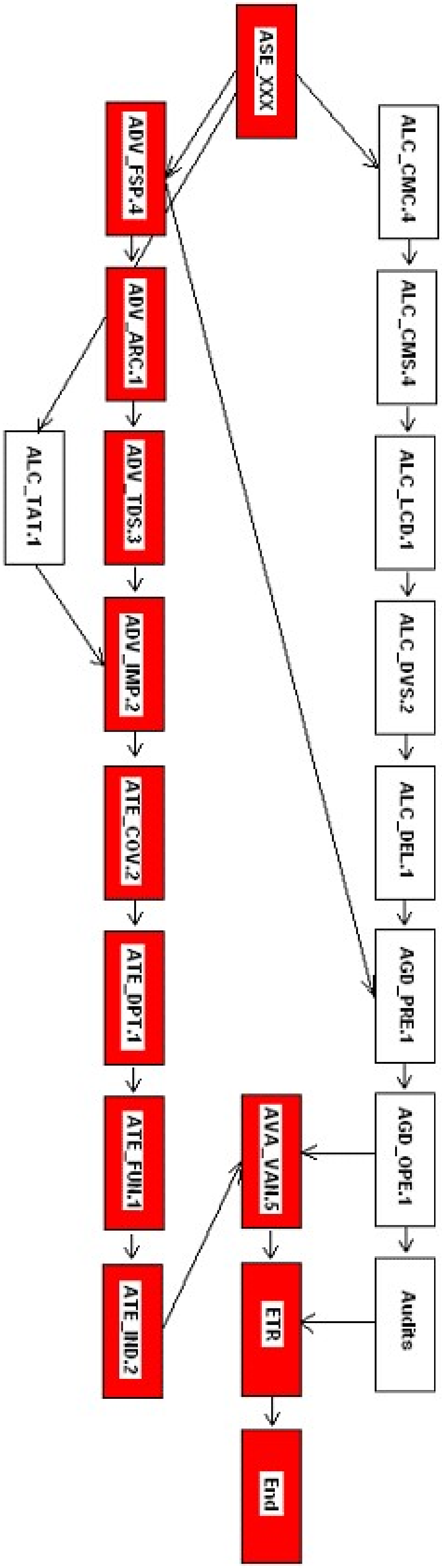
- Developers are trained to CC evaluation (reduced the number of iteration);
- The type of application is already known by the evaluators.

73 The following chapter gives a figure to illustrate an example of planning. The planning is shown as a classical sequencing of tasks. Not just the CC dependencies are considered, but also the fact that the evaluator has to gain the knowledge of the product to go through evaluation.

74 In the figure, tasks in red are the critical tasks. A delay on this critical path results in a delay of the global evaluation.

75 It is also important to note that this approach described in Annex A can be different in scenarios with site certification. The guidance supporting document CCDB-2007-11-001 “Site Certification” [SCER] defines the process, the criteria and methodology and interpretations respectively for the evaluation and certification of CC development sites. It enables the evaluation/certification of those sites in a modular fashion and without a relation to a specific TOE. The results of the site certification process can be re-used in a CC product evaluation later on e.g. in a smartcard evaluation.

A.2 Planning


ASE_XXX
ADV_FSP.4
ADV_ARC.1
ALC_TAT.1
ADV_TDS.3
ADV_IMP.2
ATE_COV.2
ATE_DPT.1
ATE_FUN.1
ATE_IND.2
ALC_CMC.4
ALC_CMS.4
ALC_LCD.1
ALC_DVS.2
ALC_DEL.1
AGD_PRE.1
AGD_OPE.1
AVA_VAN.5
Audits
ETR
End

76 The outlined approach above describes the basic top-down approach for a new design when you specify the security architecture and its assessment before you specify the details of the product design. For an existing design, ADV_ARC needs to be performed after ADV_TDS/IMP to be able to consider TDS/IMP details as required by CC/CEM.

Annex B Smartcard sub-processes

B.1 Introduction

77 The purpose of this annex is to identify the development sub-processes and to provide a CC oriented methodology for each sub-process in that evaluation may proceed as smoothly as possible.

78 To anticipate their development capability to comply to the requirements of CC product evaluations, the developers can set up and prepare the internal development methodology in order to achieve the best chance of success with little amount of work. This minimizes the duration of the product evaluation.

79 The TOEs considered to illustrate these definitions are Embedded Software or Application Software.

80 The target evaluation level to illustrate this paper is EAL4 augmented with the components ALC_DVS.2, AVA_VAN.5.

B.2 Identification of sub-processes

81 Only the software development sub-processes for smartcard product developer and for application developer are detailed further. This could apply to "IC manufacturing process".

82 The identified sub-processes are the following:

- Development environment
- Security Target
- Guidance documentation
- Development/Test

83 The first sub-process can lead to an evaluation and certification that will be re-used in the smartcard product evaluation. For the three other sub-processes, the re-usability consists of preparing the development methodology through developers 'training and template documents' preparation.

B.3 Development environment sub-process

B.3.1 Sub-process definition

84 The following process relates to the generic methodology for the secure development of all products; a separate evaluation could be done in a maintenance mode. The following deliveries related to the assurance classes are described below:

Class	Component	Development environment Sub-process	Responsible
ALC: Life cycle support	ALC_CMC.4 Production support, acceptance procedures and automation	ALC_CMC.4	QA responsible
	ALC_CMS.4 Problem tracking CM coverage	ALC_CMS.4	QA responsible
	ALC_DEL.1 Delivery procedures	ALC_DEL.1	Project responsible
	ALC_DVS.2 Sufficiency of security measures	ALC_DVS.2	Security responsible
	ALC_LCD.1 Developer defined life-cycle model	ALC_LCD. 1	QA responsible
	ALC_TAT.1 Well-defined development tools	(1)	(1)

(1) Those components cannot be described generically as the others because they are specific to the product type.

85 The roles of previous responsible are described in the following table:

Responsible	Role
Quality Assurance responsible	To define quality assurance procedures and verify the application of previous procedures.
Project responsible	To define delivery procedures and verify the application of previous procedures.
Security responsible	To define security procedures and verify the application of previous procedures.

86 In the development environment process, the product development follows a development life cycle model with the description of development/acceptance/maintenance general process and the mandatory documents as requirement specification, configuration management plan, functional requirement, high-level design, validation test plan, validation test specification, validation test result, anomaly list report.

87 In the development environment process, what is needed for the product development evaluation is used for all product development.

88 Note: development environment sub-process documentation could apply to card issuance, too.

B.3.2 Generic methodology

89 The generic methodology could be the following:

- Train the developer to the CC assurance components.
- Prepare a template document to ALC_CMC.4, ALC_CMS.4, ALC_DEL.1, ALC_DVS.2 and ALC_LCD.1 requirements and reference related documents as configuration management plan, configuration list, an acceptance plan, the delivery and security procedures.
- Complete each “reference” document according to the TOE specificities.

B.3.3 Development environment process evaluation

90 The possible target should be a system that could address the software development environment process for smartcard software.

91 The life cycle phase concerned is limited to software specification, implementation and testing with the delivery to IC manufacturer.

92 The possible objectives requested for a smartcard product development evaluation:

- To assure confidentiality, integrity and authorized access to the products, documentation and development tools (ALC_DVS.2, ALC_LCD.1),
- To assure only authorized modification of development procedures, tools and related documents (ALC_TAT, ALC_DVS.2, ALC_LCD.1, ALC_CMC.4, ALC_CMS.4),

- To track all the TOE representation evolutions (documentation, implementation, tests, tools) in a configuration management for the product development (ALC_CMC.4, ALC_CMS.4),
- To protect the deliveries against any modification or disclosure during the delivery process (ALC_DEL.1).

93 Note: the development environment process evaluation could replace the generic methodology steps 1 and 2.

B.3.4 Smartcard product evaluation based on development environment process evaluation

94 The smartcard product evaluation uses the development environment process evaluation results in order to answer to ALC_TAT, ALC_CMC.4, ALC_CMS.4, ALC_DEL.1, ALC_DVS.2 and ALC_LCD.1 security assurance requirements.

95 Some specific refinements of the components specific to the considered product will then need to be provided such as the configuration list.

96 The product evaluation could use, when available, the development environment process evaluation maintenance certificate in order to replace the audit.

B.4 Security target sub-process

B.4.1 Sub-process definition

97 The Security Target document is the basis of the evaluation. It is the representation of an identified TOE with set of security functions and assurance measures specifications to address an identified set of security requirements which themselves address an identified set of security objectives.

98 In order to apply the Security Target to the product range, the developer should produce a generic security target to optimize cost and time.

Class	Component	Security Target Sub-process	Responsible
ASE: Security Target evaluation	ASE_SPD.1 Security problem definition	ASE_SPD.1	Project responsible
	ASE_INT.1 ST introduction	ASE_INT.1	Project responsible
	ASE_OBJ.2 Security objectives	ASE_OBJ.2	Project responsible
	ASE_CCL.1 Conformance claims	ASE_CCL.1	Project responsible
	ASE_REQ.2 Derived security requirements	ASE_REQ.2	Project responsible

	ASE_ECD.1 Extended components definition	ASE_ECD.1	Project responsible
	ASE_TSS.2 TOE summary specification with architectural design summary	ASE_TSS.2	Project responsible

99 The roles of previous responsible are described in the following table:

Responsible	Role
Project responsible	To describe how the TOE functionality address an identified set of security requirements which themselves address an identified set of security objectives.

B.4.2 Generic methodology

100 The generic methodology could be the following:

- Train the developer to the CC introduction (CC Part 1), security functional requirements (CC Part 2), and the security assurance requirements (CC Part 3).
- Prepare a template document that answer to ASE_XXX.n requirements.
- Complete each “reference” part according to the TOE specificity.

B.5 Guidance documentation sub-process

B.5.1 Sub-process definition

101 In order to ensure the development capability complies with the targeted evaluation assurance level, the developers should prepare, complete and maintain the related documents according to CC component requirements.

102 The reference document for all documents is the Security Target. Moreover, the production of the TOE is necessary for the evaluation.

Class	Component	Guidance documentation Sub-process	Responsible
ALC: Life cycle support	ALC_DEL.1 Delivery procedures	ALC_DEL.1	QA responsible

	ALC_CMC.4 Production support, acceptance procedures and automation	ALC_CMC.4	QA responsible
AGD : Guidance documents	AGD_PRE.1 Preparative procedures	AGD_PRE.1	Documentation responsible
	AGD_OPE.1 Operational user guidance	AGD_OPE.1	Documentation responsible

103 The roles of previous responsible are described in the following table:

Responsible	Role
Documentation responsible	To define documentation for administrator and user.
Quality Assurance responsible	To define quality assurance procedures and verify the application of previous procedures.

104 A generic methodology is necessary to prepare, complete and maintain the related documents. In order to reduce time and cost, it is very important to train developers in the generic methodology. Then, the reuse can be applied to the product range.

105 Note: this guidance documentation sub-process could apply to card issuer, too.

B.5.2 Generic methodology

106 The generic methodology could be the following:

- Train the developer in the CC assurance components.
- Prepare a template document that answers AGD, and ALC_CMC.4 requirements and reference related documents as administrator/user guidance, and preparation processes. All these documents are dependent of ADV_FSP.4 requirements.
- Complete each “reference” document according to the TOE specifics.

B.6 Development / Tests sub-process

B.6.1 Sub-process definition

107 In order to anticipate the development capability to comply with the targeting evaluation assurance level, the developers should prepare, complete and maintain the related documents according to CC component requirements.

108 The reference document for all documents is the Security Target. Moreover, the production of the TOE is necessary for the evaluation.

Class	Component	Development/Tests Sub-process	Responsible

ADV: Development	ADV_FSP.4 Complete functional specification	ADV_FSP.4	Developer
	ADV_ARC.1 Security architecture description	ADV_ARC.1	Developer
	ADV_IMP.1 Implementation representation of the TSF	ADV_IMP.1	Developer
	ADV_TDS.3 Basic modular design	ADV_TDS.3	Developer
ALC: Life cycle support	ALC_DVS.2 Sufficiency of security measures	ALC_DVS.2	Project responsible
	ALC_LCD.1 Developer defined life-cycle model	ALC_LCD.1	Project responsible
	ALC_TAT.1 Well-defined development tools	ALC_TAT.1	Project responsible
ATE: Tests	ATE_COV.2 Analysis of coverage	ATE_COV.2	Qualifier
	ATE_DPT.1 Testing: basic testing	ATE_DPT.1	Qualifier
	ATE_FUN.1 Functional testing	ATE_FUN.1	Qualifier
	ATE_IND.2 Independent testing - sample		

109 The roles of previous responsible are described in the following table:

Responsible	Role
Developer	To develop the Embedded Software or the Application.
Project responsible	To define the security policy and development tools.
Qualifier	To test the Embedded Software or the Application.

110 A generic methodology is necessary to prepare, complete and maintain the related documents.

111 In order to reduce time and cost, it is very important to train developers to the generic

methodology. Then, the re-usability can apply to the product range.

B.6.2 Generic methodology

112 The generic methodology could be the following:

- Train the developers to the CC assurance components.
- Prepare a template document that answer to ALC_TAT.1 requirements and reference related documents as development procedures, development tools etc.
- Prepare a template document that answer to ADV requirements and reference related documents as functional specification, design etc.
- Prepare a template document that answer to ATE requirements and reference related documents as validation test plans, validation test specifications, validation test results. All these documents are dependent on ADV requirements.
- Complete each “reference” document according to the TOE specification.

Annex C Penetration Testing methodology

C.1 Objective

113 The aim of this annex is to describe the methods used by security evaluation laboratories to define the attacks and strategies list of a smartcard security evaluation. This method shall be independent of the Common Criteria AVA_VAN component. The overall methodology is the same for all the smartcard security evaluations (hardware and software evaluations). Even if the examples given here are issued from “IC with embedded software” evaluations, the method should be the same for all types of evaluations.

C.2 List of potential vulnerabilities

114 In Security Certification Schemes, accredited laboratories on smartcards need to show competences in this area (attacks, vulnerabilities). This background is the starting point, the list of what can be applied for attacking a smartcard is called: Potential vulnerability list.

115 Additional guidance on attack methods should be provided by National Schemes to accredited laboratories in the field of smartcards and similar devices. This information is also supported by the document “JIL Application of Attack Potential to Smartcards” [APSC].

116 General requirements (knowledge, skills, standard and bespoke equipment, etc.) for accredited laboratories executing IC or composite evaluations are described in the document “Minimum ITSEF Requirements for Security Evaluations of Smart cards and similar devices” [MIRSE].

C.3 Defining the penetration tests

117 During the evaluation, each CC task will modify the initial list in:

C.3.1 Removing attacks

118 The vulnerability analysis of the product (and mainly of the source code) can remove some potential attacks from the list of penetration tests if the countermeasures implemented by the developers are judged adequate by the evaluator, for example anti DFA implementations could be judged as efficient for protecting against DFA.

119 Remarks: the judgment is done by the evaluator. In case of doubt, the attack is always left in the list. For well known attacks the evaluator will need to justify that the attack is not tried and this justification will must be agreed by the certification body.

C.3.2 Adding attacks

120 Some configurations in the products could give ideas to the evaluator for using observation or perturbations means to gain secrets of the card. Such an approach is called an attack strategy and is inserted in the list.

121 For example, if the code implements a single test before giving critical information and if a perturbation is known with the effect of inverting a test, the evaluator will try to define

a strategy for exploiting this potential vulnerability.

122 A strategy is different from an attack path because it is incomplete. For example, in the previous case, an attack path will need to include information about the synchronization in applying the perturbation, this information could come later in the evaluation or in the tests themselves.

123 Remarks: in this step, all the CC tasks of the evaluation could modify the list (weaknesses of the protocol can be detected in analysing the specifications, the guidance or the AVA_VAN tasks), but the major contribution to the list is done by the ADV tasks and mainly by the ADV_IMP task.

124 Different AVA_VAN levels have different attack potential levels associated, and an important difference between them is the vulnerability analysis method. They go from vulnerability survey, vulnerability analysis, focused vulnerability analysis, methodical vulnerability analysis to advanced methodical vulnerability analysis.

C.3.3 Tuning existing attacks

125 Some attacks are known from a "generic" point of view. Specific strategies, specific parameters need to be "tuned" depending on the application (IC and software) characteristics. The analysis of the application (Source code, IC) will help the evaluator in giving some information to tune the attacks.

126 For example, DPA can be done on various place of a DES, DPA is done by acquiring curves, processing them to extract "interesting" characteristics and exploiting them, depending on the IC characteristics signal processing methods need to be adapted.

127 The method of defining the test list is summarized in the following diagram:

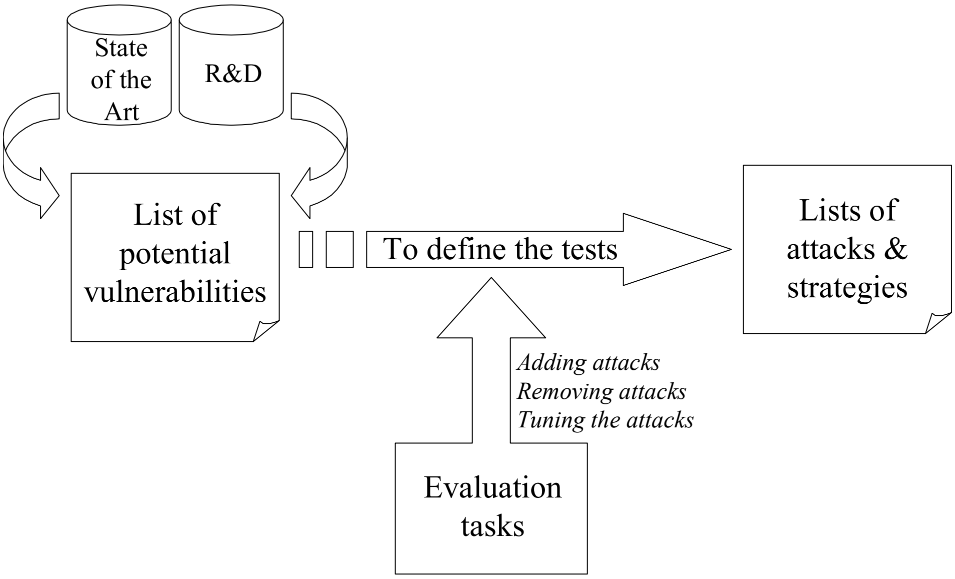


Figure 7 – Method of defining the penetration test list of attacks to the Smartcard

C.4 List of attacks and strategies

128 This is the final list of the tests that will be performed by the evaluator. It includes attacks (when the attack path is fully defined) or strategies (when the attack path is not fully defined and when some questions are still to be answered before implementing the attack).

129 At this point, the quotation of the attacks are taken into account. Attacks with an attack path driving to a quotation higher than the targeted level could be suppressed (in accordance with the Certification Body), but the corresponding vulnerabilities are identified as «residual vulnerabilities».

130 Finally, the tests are done, attacks that succeeded are quoted:

- If a successful attack is quoted lower or equal to the targeted level, the task is FAIL.
- If a successful attack is quoted higher than the targeted level, it will become a «residual vulnerability».

C.5 Conclusions

131 The points we would like to point out are the following:

- Evaluating smartcards requires a laboratory experienced in performing penetration testing, with a good knowledge of the state of the art (not always published) and the capability of doing new attack research.
- Penetration testing is not just applying known attacks on the product. The ADV tasks are fundamental to identify potential vulnerabilities. The ADV_ARC.1 and ADV_IMP.1 tasks are the key tasks in analysing the TOE.
- The difference between black box testing and AVA_VAN.5 is not just time or testing effort. AVA_VAN.5 uses the knowledge gained in the ADV tasks and the attack strategy could be very different from what is done in a black box evaluation. AVA_VAN.5 involves performing penetration tests with high attack potential in terms of CEM.