Supporting Document
Mandatory Technical Document

Composite product evaluation for
Smart Cards and similar devices

April 2012

Version 1.2

CCDB-2012-04-001

Foreword

This is a supporting document, intended to complement the Common Criteria version 2 and 3 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: NLNCSA

Document History:
V1.2, April 2012: Update of platform certificate validity rules for composition
V1.0, September 2007 : Initial release.

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:	*Agence Nationale de la Sécurité des Systèmes d'Information*
Germany:	*Bundesamt für Sicherheit in der Informationstechnik*
Italy :	*Organismo di Certificazione della Sicurezza Informatica*
Netherlands:	*Netherlands National Communications Security Agency*
Spain:	*Ministerio de Administraciones Públicas and Centro Criptológico Nacional*
United Kingdom:	*Communications-Electronics Security Group(CESG)*

They also acknowledge the contribution of the work done by several smart card vendors, evaluation labs, and other companies organised within:

- *eEurope*
- *International Security Certification Initiative (ISCI)*

Table of contents

Figures

Tables

1 Introduction

1.1 History

1 The Common Criteria (CC) are being widely used for smart card products security evaluation. Smart card evaluation showed very early a need for interpretation and supporting documents.

2 The initial reason was that a smart card is built up with a combination of two parts: a hardware integrated circuit part and a software part often developed by different actors with specific objectives.

3 One objective was to independently perform one evaluation of a platform to address several applications and customers.

4 Another objective was to create one or several applications to load on one or several certified platforms.

5 The objective for Application Integration was to install one or several applications onto one already certified platform to reduce the evaluation effort keeping a high level of confidence.

6 To achieve these objectives, a transfer of knowledge and a reuse of evidences have been defined.

1.2 Definitions

7 The hardware part and associated libraries (if applicable) is evaluated independently as it can be used with many different software applications.

8 The software is embedded in the hardware and is built to operate with this hardware. The resulting product is the one which is used in the field (cellular phones, banking cards, health cards, Identity, digital signature, e-pass, e-ticketing etc.) and on which customers/users need to gain confidence.

9 Another specificity of the smart card type product is that the software part has to use, control, configure or activate the security mechanisms provided by the hardware.

1.3 Composite product evaluation and ACO (CC V3)

10 Although the CC version 3 introduces the specific assurance class ACO for composition, this class is not suitable for usual smart card and similar devices evaluation.

11 ACO addresses a TOE composed of two certified TOEs: the Base TOE and the Dependent TOE (see Figure 1). The evaluation of the composed TOE consists in evaluating the interaction between both TOEs, reusing evaluation results of Base TOE and Dependent TOE.

12 The result of this evaluation is not an EAL level, but a CAP level which is not comparable to EAL level. Furthermore, ACO class is applicable up to Extended-Basic assurance level, whereas smart cards especially in banking or signature type application require 'High Level' assurance.

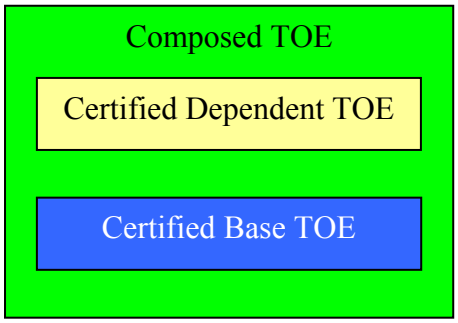


Figure 1 - ACO composed TOE (package CAP)

13 For smart card and similar devices the composite product is the final product for which **an EAL level certification is required**. This allows a direct comparison with similar products certified after a single evaluation.

14 Considering smart card architecture, it is composed of a hardware platform and a software application. In the **Composite TOE** evaluation, the platform is certified, the application is evaluated and the results of the platform certification are reused. See Figure 2 for security certification of the entire Composite TOE.

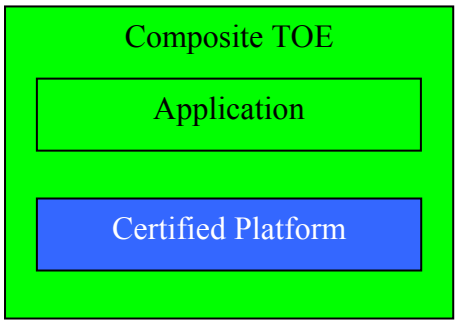


Figure 2 - Composite product evaluation (current approach)

15 The hardware platform has no 'strictly functional' properties related to security. It provides mechanisms to protect the composite product assets, but the composite product behaviour depends widely on the software application having to use, to configure and activate these security mechanisms.

16 Therefore, the hardware platform evaluation results provide security recommendations and conditions for the software application implementation. The composite product evaluator shall examine amongst other that the combination of both products does not lead to any exploitable vulnerability.

17 The smart card composite evaluation methodology defines precise work units with clear statement on the information needed from the platform developer and provides an agreed "framework" for information transfer from platform to composite product evaluator.

18 The information required is already available from the platform evaluation tasks and no additional work is required from platform developer.

- There is no need for details on the platform development class ADV.
- The user guidance (AGD) of the platform is considered early in the development of the composite product and provides all interfaces information needed.
- The evaluated interfaces of the platform are relied upon.
- All relevant interfaces between platform and application are in the scope of the composite product evaluation.
- Test (ATE) and vulnerability assessment (AVA) are performed on the composite product taking advantage of platform evaluation results.

19 Current concept of the Composite TOE evaluation does not limit the composite evaluation in EAL and resistance against attacks, i.e. up to 'high', whereas Composed TOE (CAP package) is limited by resistance against attacks 'extended-basic'.

1.4 Objective and scope

20 The objective of this document is to precisely define tasks for the different parties involved in the composite product evaluation.

21 The aim is not to define an additional assurance class, but to define refinements to the existing assurance requirements for a composite product evaluation.

22 This document does not address smart cards only, but any other security IT technology where an independently evaluated product is part of a final composite product to be evaluated.

23 Therefore this document addresses **smart cards and similar devices**.

2 Definitions / Terminology

2.1 Definitions

24 The *composite product* is a product consisting of at least two different parts, whereby one of them represents a single product having already been evaluated and certified.

25 The *composite TOE* is defined in such a way that it comprises the whole composite product, i.e. the certified product is declared to be part of the composite TOE: The composite product is equivalent to the composite TOE.

- An evaluation of the composite TOE is a *composite evaluation*.
- Usually a composite product consists of two components, whereby the first one represents an '*underlying platform*'[1] and the second one constitutes an '*application*' running on this platform. The underlying platform is the part of the composite product having already been evaluated (see Figure 3):

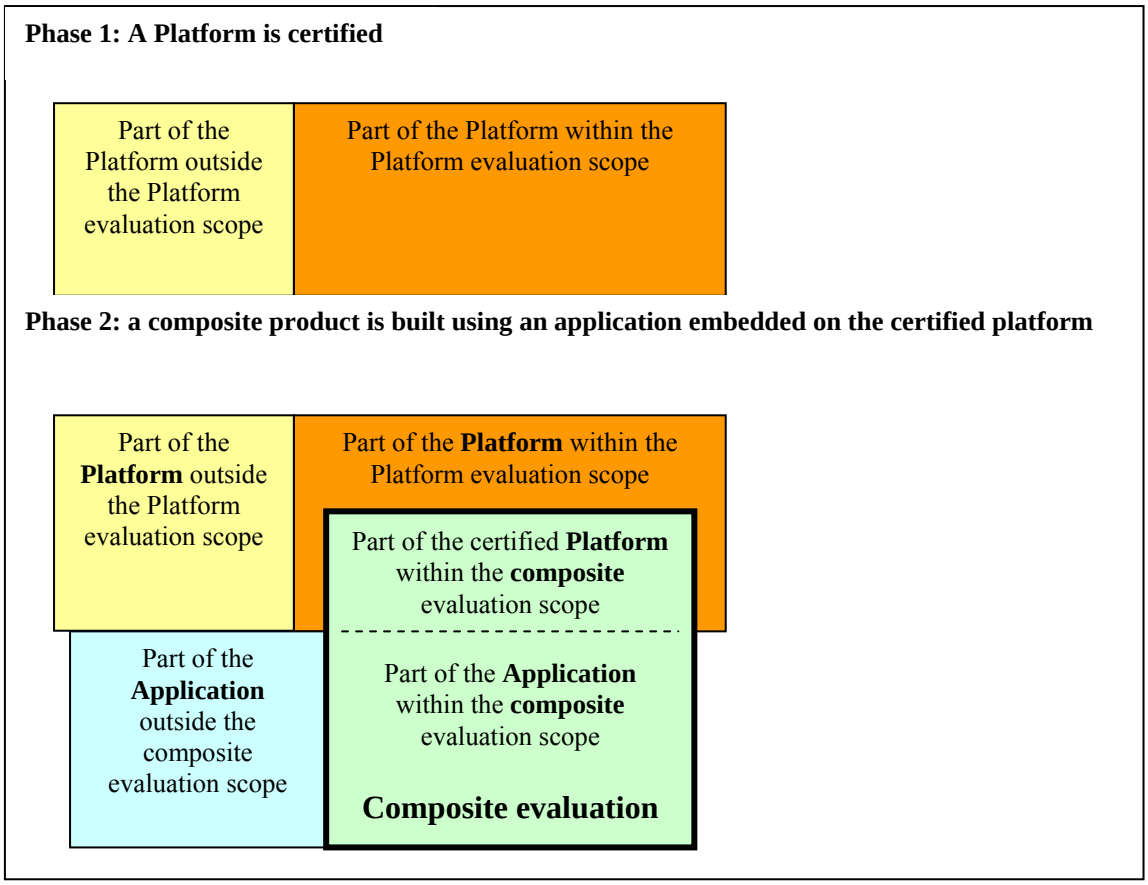


Figure 3 - Composite evaluation scope example

- Exemplifying we can mention an operating system ('application') running on a hardware platform ('underlying platform') or a JavaCard™ applet ('application') running on a Java Card runtime environment ('platform').

[1] such a platform might also be called 'abstract machine' or 'virtual machine'

Further examples are crypto-boxes and secure terminals containing a SAM/HSM[2]: the crypto-box hardware with a boot-loader (and perhaps with a core operating system) represents the 'underlying platform', a special crypto-box application running on it is the 'application'; the evaluated SAM/HSM plays usually the *server role* and represents the 'underlying platform', the SAM-external terminal application software playing the SAM-*client-role* represents the 'application'.

- In order to keep the document and terminology simple we consider only the two-component composite products and use the term '*platform*' for the underlying certified product and the term '*application*' for the software product running on the platform.

26 These definitions comply with ACO class definitions where:

- A platform is the base component,
- An application is the dependent component.

2.2 Roles

27 The following roles shall be considered in the composite evaluation activities:

- **Platform Developer**: Entity developing the platform; it might also be the sponsor of the platform evaluation.
- **Platform Evaluator**: Entity performing the platform evaluation.
- **Platform Certification Body**: Entity performing the platform certification, defined in CC V3 terminology as evaluation authority.
- **Application Developer**: Entity developing the application running on the platform.
- **Composite Product Integrator**: Entity installing the applications on the platform.
- **Composite Product Evaluator**: Entity performing the composite product evaluation.
- **Composite Product Certification Body**: Entity performing the composite product certification defined in CC V3 terminology as evaluation authority.
- **Composite Product Evaluation Sponsor**: Entity in charge of contracting the composite product evaluation.

28 Each evaluation shall associate particular organizations or persons to these generic roles.

29 In order to illustrate the role of the **Composite Product Integrator** let us exemplify:

[2] security access module / hardware security module

- Smart cards: The ‘underlying platform’ is an integrated circuit and the **Platform Developer** is the integrated circuit (chip) manufacturer; the ‘application’ is a card operating system and the **Application Developer** is the developer of the smart card software. In this case, the role of the Composite Product Integrator is played by (i) the chip manufacturer embedding the core of the operating system into the ROM of the chip, then by (ii) the card manufacturer usually loading some parts of the operating system and the applications into the EEPROM of the chip.

- Java Card technology-enabled devices: The ‘underlying platform’ is the Java Card runtime Environment (Java Card RE) on chip and the **Platform Developer** is the card manufacturer/issuer; the ‘application’ is the Java Card applet. In this case, the role of the **Composite Product Integrator** is played by the domain/application service provider or by a trust centre loading the applet and often personalizing the card electronically.

- Crypto-Boxes: The role of the **Composite Product Integrator** is often played by the crypto-box manufacturer itself: he produces the entire crypto-box including the operating system and the concrete applications and then delivers the final composite product to a customer. In general, the customizer of the crypto-box is the Composite Product Integrator.

3 Composite evaluation concept

3.1 What are the issues?

30 The assets to be protected are the final composite product assets defined in the composite product Security Target.

31 The security mechanisms involved in the protection of these assets are those provided by the platform and by the application itself.

32 Some of the security mechanisms provided by the platform may require configuration, programming or activation by the application.

33 Therefore the **Application Developer** needs all the information (in form of a guidance or user's manual) related to the platform security mechanisms the application has to manage.

34 Furthermore he needs the platform security target in order to build the composite product security target and to ensure consistency of security definition between both developments. Evaluation is performed and validated on the final composite product.

35 The **Composite Product Evaluator** has to examine, whether the level of security required by the Security Target of the resulting composite product is achieved, when both parts are combined. Therefore the **Composite Product Evaluator** has to execute the evaluation tasks needed with respect to the Security Target of the final composite product and to provide the related ETR. In this perspective, the **Composite Product Evaluator** should reuse the platform's evaluation and certification result thus saving cost and time.

3.2 What information is needed?

36 The **Composite Product Evaluator** does not need all platform evaluation results. The security certificate and the certification report assure that the platform has been evaluated according to the Common Criteria. The **Composite Product Evaluator** will need complementary information on the assurance measures where both developments interfere. The **Composite Product Evaluator** will need the same level of knowledge about the platform as the **Application Developer** to check that the application meets the security recommendations on the platform usage. In addition to the standard amount of evaluation contributions according to the assurance package chosen for the composite evaluation (e.g. an EAL level) evaluation, he will need the following (see section 4.7 'Deliveries' for further details):

- All the information delivered from the **Platform Developer** to the **Composite Product Integrator**,
- All the information delivered from the **Platform Developer** to the **Application Developer**,

- ETR for composite evaluation prepared by the **Platform Evaluator**, see chapter 5 'ETR for composite evaluation' (including information about vulnerability analysis and penetration testing),

- Information on compliance of the Security Targets and the designs of the platform and the application prepared by the **Application Developer**,

- Information on compliance of the delivery procedures of the **Platform** and **Application Developers** with the acceptance procedure of the **Composite Product Integrator**, and

- Information on integration of both parts using their correct certified versions and the correct configuration parameters. This information shall be prepared by the **Composite Product Integrator**; it also implies assurance that the application is correctly managed by the **Platform Developer** (e.g. in the case of smart card where ROM code is supplied for masking on the platform).

37 **Composite Product Certification Body** will need the same amount of information as the **Composite Product Evaluator**.

3.3 Case of composite product change

38 In case of composite product changes due to a change of the platform or the application or both, please refer to [CC AC].

39 If a change comes from the platform, the assessment of the change for the platform is given by the **Platform Certification Body**. On this basis, the assessment of the change for the composite product is given by the **Composite Product Certification Body**.

40 If a change comes from the application, the assessment of the change for the composite product is given by the **Composite Product Certification Body**.

3.4 Specific case when the application is already certified

41 In the case where both platform and application have already been certified, a partial evaluation work may be performed regarding the results already obtained from previous application evaluation. Nevertheless, the composite evaluation tasks as defined in this document are still required.

4 Composite evaluation activities description

42 The current approach can be applied independent of the evaluation assurance level (EAL) for the composite product aimed. Where some evaluation activities are not applicable due to the EAL chosen, they are also not expected to be applied.

43 For the following paragraphs, we assume that the level of assurance of the platform is equivalent or higher compared to the composite product evaluation level.

44 Other cases must be discussed within the schemes.

45 The composite-specific developer and evaluator action elements as well as the evaluator actions (work units) belonging to the composition activities are defined as the refinements for composite evaluation, see Appendix 1: Composite-specific requirements.

4.1 Evaluation of the composite product Security Target

46 A Security Target for the composite product has to be written and evaluated.

47 The **Composite Product Evaluator** has to examine that the Security Target of the composite product[3] does not contradict the Security Target of the underlying platform[4]. In particular, it means that the **Composite Product Evaluator** has to examine the Composite- and the Platform- Security Target for any conflicting assumptions, compatibility of security objectives, security requirements and security functionality needed by the application.

[R1] This task can be reduced, if some matching has been checked for Protection Profiles claimed by each Security Target.

[R2] The **Composite Product Evaluation Sponsor** must ensure that the security target of the platform is available to the **Application Developer**, to the **Composite Product Evaluator** and to the **Composite Product Certification Body**. The information available in public version of the security target may not be sufficient.

4.2 Integration of the application in the configuration management system

[R3] The **Composite Product Evaluator** shall verify that the evaluated version of the application has been installed onto / embedded into the evaluated version of the underlying platform.

[R4] The **Composite Product Evaluation Sponsor** must ensure that appropriate evidence generated by the **Composite Product Integrator** is available to the **Composite Product Evaluator**. This evidence may include, amongst other,

[3] denoted by Composite-ST in the following

[4] denoted by Platform-ST in the following

the configuration list of the **Platform Developer** provided within its acknowledgement statement.

4.3 Compatibility check for delivery and acceptance procedures

[R5] The **Composite Product Evaluator** shall verify that delivery procedures of the **Application** and **Platform Developers** are compatible with the acceptance procedure used by the **Composite Product Integrator**.

[R6] The **Composite Product Evaluator** shall verify that all configuration parameters prescribed by the **Application** and **Platform Developers** (e.g. pre-personalization data, pre-personalisation scripts) are used by the **Composite Product Integrator**.

[R7] The **Composite Product Evaluation Sponsor** must ensure that appropriate evidences generated by the **Composite Product Integrator** are available to the **Composite Product Evaluator**. These evidences may include, amongst other, the

- Element of evidence for the application reception, acceptance and parameterisation by the **Platform Developer** (in form of acknowledgement statement).

4.4 Compliance of designs

[R8] The **Composite Product Evaluator** shall verify that stipulations for the **Application Developer** imposed by the **Platform Developer** in its certified user guidance and referenced in the platform certification report are fulfilled by the composite product, i.e. have been taken into account by the **Application Developer**.

[R9] The **Composite Product Evaluation Sponsor** must ensure that the following are made available to the **Composite Product Evaluator**:

- The platform-related user guidance,
- ETR for Composition prepared by the **Platform Evaluator**, see chapter 5 ‘ETR for composite evaluation’,
- The Certification Report for the platform prepared by the **Platform Certification Body**,
- A rationale for secure composite product implementation including evidences prepared by the **Application Developer**.

4.5 Composite product functional testing

[R10] Some application functionality testing can only be performed on emulators, before its embedding/integration onto the platform, as effectiveness of this testing (pass/fail) may not be visible using the interfaces of the composite product. Nevertheless, functional testing of the composite product shall be

performed also on composite product samples according to description of the security functions of the composite TOE and using the standard approach as required by the relevant assurance class. No additional developer's action is required here.

[R11] The **Composite Product Evaluator** shall check the minimal amount of the testing necessary for the current composite evaluation having been performed on the composite product as a whole. That is what is called further in the document integration testing.

[R12] Since the amount, the coverage and the depth of the functional tests of the platform have already been validated by the platform certificate, it is not necessary to re-perform these tasks in the composite evaluation. Please note that ETR for Composition (see chapter 5 'ETR for composite evaluation') does not provide any information on functional testing for the platform.

[R13] The **Composite Product Evaluation Sponsor** must ensure that the following is available to the **Composite Product Evaluator**:

- Composite product samples suitable for testing.

4.6 Composite product vulnerability analysis

[R14] The **Composite Product Evaluator** shall perform a vulnerability analysis for the composite product using, amongst other, the results of the platform evaluation and certification. This vulnerability analysis shall be confirmed by penetration testing.

[R15] In special cases, the vulnerability analysis and the definition of attacks might be difficult, need considerable time and require extensive pre-testing, if only documentation is available. The platform may also be used in a way that was not foreseen by the **Platform Developer** and **Platform Evaluator**, or the **Application Developer** may not have followed the stipulations provided with the platform certification. Different possibilities exist to shorten composite vulnerability analysis in such cases:

- The **Composite Product Evaluator** can consult the **Platform Evaluator** and draw on his experience gained during the platform evaluation.

- Separation of vulnerabilities of application and platform with the use of "open samples" ("open samples" are samples of the platform on which the **Composite Product Evaluator** can load software on his own discretion). The intention is to use test software without the application countermeasures without deactivating any platform inherent countermeasure. The aim is clearly not to repeat the platform evaluation. (Refer to [JIL AP] for further details).

[R16] The **Composite Product Evaluation Sponsor** must ensure that the following are made available to the **Composite Product Evaluator**:

- *The ETR for Composition* (ETR_COMP) prepared by the **Platform Evaluator**, see chapter 5 'ETR for composite evaluation' below, and
- The Certification Report for the platform prepared by the **Platform Certification Body**.

4.7 Deliveries

48 The tables below summarize the documentation deliveries that are exchanged between parties to enable the composite evaluation activities as defined in the previous paragraphs.

49 The **Composite Product Evaluation Sponsor** is in charge of the initialization of the process.

50 The **Composite Product Evaluation Sponsor** is responsible for maintaining or creating any **Non Disclosure Agreement** (NDA) that would be necessary between all the parties involved in the composition activities.

51 The **Non Disclosure Agreement** should be established according to the sensitivity and ownership of the information to be exchanged

##	Document / Contribution	Description
1	Platform Security Target	Security Target of the platform as referenced in the platform certification report.
2	Platform open samples for testing	Platform samples as defined in [JIL AP] Chapter 3.8.
3	Platform user guidance	It encompasses all platform user guidance and manuals needed for the **Application Developer** and the **Composite Product Integrator** being referenced in the platform certification report.
4	Platform ETR_COMP	ETR for composition as defined in chapter 5 and referenced in the platform certification report.
5	Platform certification report	Platform certification report issued by authorized **Platform Certification Bod**y.
6	Design compliance evidence	It enfolds evidence elements on how the requirements on the application design, imposed by the platform's guidance and certification report, are fulfilled in the composite product. If such a requirement was not followed, a rationale that the chosen composite product implementation is still secure shall be given here.

##	Document / Contribution	Description
7	Composite configuration evidence	It comprises (i) Identification elements of the composite product - proving that the correct, certified version of the platform is used in the composite product, - proving that the correct, evaluated version of the application has been integrated; and (ii) Evidence elements that configuration parameters prescribed by the **Platform** and **Application Developers** are actually being used by the **Composite Product Integrator**.
8	Delivery and acceptance procedures evidence	Evidence elements how the delivery procedures of the **Platform** and **Application Developers** are compatible with the acceptance procedure of the **Composite Product Integrator**

Table 1 - Definition of composition documents

52 The following table shows which documents/contributions of Table 1 shall be provided to which actor within the composite evaluation process:

		Actors				
##	Documents/contributions having to be provided to	Composite product evaluation Sponsor	Composite product integrator	Applica-tion developer	Composite product Evaluator	Composite product Certifica-tion Body
1	Platform Security Target	No	No	Yes	Yes	Yes
2	Platform open samples[5]	No	No	No	Yes	No
3	Platform user guidance	No	Yes	Yes	Yes	Yes
4	Platform ETR_COMP	No	No	No	Yes	Yes
5	Platform certification report	Yes	Yes	Yes	Yes	Yes
6	Design compliance evidence	No	No	No	Yes	Yes
7	Composite configuration evidence	No	No	No	Yes	Yes
8	Delivery and acceptance procedures evidence	No	No	No	Yes	Yes

Table 2 - Main Deliveries between actors

[5] Only if requested by composite product evaluator as defined in [CC-AP]

53 The next table shows some example of Composite TOE use cases with definition of the components and the roles.

	Composite TOE example		
Components & roles definitions	**Smartcard –I** The composite TOE is built of - a Security IC with an application code loaded in ROM (Masking operation) and application data loaded in EEPROM.	**Smartcard –II** The composite TOE is built of - a Security IC without ROM, but offering Flash technology and Flash loader - an application code and data loaded into the flash by a smart Card manufacturer	**Java Card** The composite TOE is built of - a Java Card Platform - a Java card application: the applet
The **Platform** is	The Security IC	The Security IC with the Flash memory and the Flash Loader	The JavaCard Platform including Card Manager with Applet loader facility
The **Application** is	The Operating System code plus additional data files	The Operating System code, Flash memory initialization data and application data	The Applet
The **Platform Developer** is	The Security IC Manufacturer: - Develops and manufactures the Security IC	The Security IC Manufacturer: - Develops, manufactures and delivers the Security IC with Flash technology to the Composite Product Integrator	The Java Card Platform developer: - Develops the Java Card with applet loading mechanism to the Composite Product Integrator.
The **Application Developer** is	The Smartcard Software developer: - Develops the application; - Provides the application to Composite product integrator	The Smartcard Software developer: - Develops the application; - Delivers the application to the Composite Product Integrator	The Applet developer: - Develops the applet; - Delivers the applet to the Composite Product Integrator
The **Composite Product Integrator** is	The Security IC Manufacturer: - is in charge of OS masking in ROM and of loading Application data in EEPROM; - Delivers the Composite TOE to be evaluated	The Card Manufacturer: - is in charge of loading the application into the flash using Security IC flash loader; - Delivers the Composite TOE to be evaluated	The Card Issuer: - Loads the applet on the Java Card platform using applet loading mechanism; - Delivers the Composite TOE to be evaluated

Table 3 - Example of composite TOE use cases

5 ETR for composite evaluation

5.1 Objective of the document

54 A standard Evaluation Technical Report (ETR) contains proprietary information that cannot be made public. The *ETR for composite evaluation* (ETR_COMP) document is compiled from the ETR in order to provide sufficient information for composite product evaluation with a certified platform. It should enable the **Composite Product Evaluator** and the respective **Certification Body** to understand the considered attack paths, the performed tests and the effectiveness of countermeasures implemented by the platform.

55 A template for an **ETR_COMP** document is given in Appendix 2: ETR for composite evaluation template.

5.2 Generic rules:

[R17] The *ETR for composite evaluation* should be produced by the **Platform Evaluator** based on the platform evaluation results. This task should be considered when determining the evaluation work program to reduce additional cost and effort.

[R18] The content of ETR_COMP has to strike the right balance between protecting platform developer's and/or **Platform Evaluator's** proprietary information and providing sufficient information for the **Composite Product Evaluator** and the respective **Certification Body**, cf. Table 2 above.

[R19] ETR_COMP shall not include information affecting national security.

[R20] The information provided must be approved by all parties involved in the platform evaluation (i.e. the Evaluator, the Certification Body, the developer and sponsor of the evaluation). The platform Certification Body shall validate its consistency with the original ETR. The platform certification report shall reference the *ETR for composite evaluation*.

[R21] If the current ETR_COMP itself relies on a composite evaluation, and if there is direct interface with the previous platform, the reference to this previous composite evaluation ETR_COMP must be supplied.

5.3 Content of the ETR for composite evaluation

[R22] The information required is focused on:

a) Formal information about the platform like its exact identification, reference to the certification report etc.

b) Information about the Platform design.

c) Information about the evaluated configuration of the Platform.

d) Information on delivery procedures and data exchange.

e) Information about penetration testing of the Platform including the considered attack paths and summary of test results.

f) Observations and recommendations for users.

5.3.1 Formal information

[R23] This section of ETR_COMP shall provide formal information on the platform evaluation as:

- product identification,
- sponsor and developer identities,
- identities of the evaluation facility and the certification body,
- assurance level of the evaluation,
- formal evaluation and certification results like pass/fail,
- references to the ETR.

5.3.2 Platform design

[R24] This section of ETR_COMP shall provide a high-level description of the IT product and its major components based on the deliverables required by the assurance class ADV of the Common Criteria. The intent of this section is to characterize the degree of architectural separation of the major components and to show possible technical dependencies between the platform and an application using the platform (e.g. dependencies between HW platform and SW application).

5.3.3 Evaluated configuration

[R25] This section of ETR_COMP shall provide information about the evaluated configuration of the Platform based on the developer's configuration list or relevant parts as needed or on a case by case basis. The platform must unambiguously be identifiable and this identification shall be commensurate with the evaluated configuration as stated in the platform certification report.

[R26] If applicable, generation and installation parameter settings being security relevant for the Platform should be explained and their effect on the defence against attacks be outlined (e.g. key length, counters limits).

[R27] Evidence about the evaluation of the configuration management coverage (CC V2.x: ACM_CAP; CC V3 ALC_CMC) can be necessary for a specific type of Platform, if it is relevant for supporting composite evaluations (e.g. evidence about integration of a SW-application in the configuration management of a combined HW/SW-production). Therefore, beside the evaluation evidence about the principle capability of the configuration management system, specific composite configuration evidence may be necessary.

More precisely, as a platform may have several evaluated configurations, one dedicated configuration list for composition shall be delivered to the Composite Product Evaluator. With this document the evaluator should be able to make the link between the evaluated platform configuration items and the composite product configuration.

An example of hardware platform and embedded OS configuration list document is shown in the table below:

Hardware platform evaluated configurations list document
Identification of the TOE:
Identification of the configuration for the ROM:
Identification of the configuration for the EEPROM:
Configuration options:
Identification of Mask Set:
Layout file:
Embedded IC software identification:
Configuration list of the composite product (Additional information to be supplied for composite evaluation)
Identification of the composite TOE:
Identification of the configuration for the ROM:
Identification of the configuration for the EEPROM:
Configuration options:
Additional information if any

Tabel 4 - Configuration list example for hardware Platform

5.3.4 Delivery procedures and data exchange

[R28] For supporting composite evaluation, evaluation evidence can be necessary for delivery of the platform, and acceptance procedures of the application and related data to be integrated during development and production. Therefore, evaluation evidence about ADO_IGS (CC V2.x) resp. AGD_PRE[6] (CC V3) and ADO_DEL (CC V2.x) resp. ALC_DEL + AGD_PRE[7] (CC V3) might be relevant.

5.3.5 Penetration Testing

[R29] This section of ETR_COMP shall provide information about the independent vulnerability analysis performed by the **Platform Evaluator** with the attack scenarios having been considered, the penetration testing having been performed and the reference to the corresponding rating (quotation) of the attack potential.

[R30] Information about penetration testing should include details necessary for understanding the attack scenarios/paths and the assessment of penetration results.

[6] [1.2C]
[7] [1.1C]

[R31] The attack scenario descriptions should provide sufficient details to reproduce attacks, which require additional countermeasures in the composite TOE.

[R32] In accordance with the requirements of CEM[8], this information is available within the ETR. So it can be compiled for ETR_COMP.

[R33] This section shall also mention the rating of access to 'open samples', if they exist (public/restricted/sensitive/critical).The use of 'open samples' shall be considered in the assessment of the attack path. Please note that 'open samples' are evaluation tools, but do not represent a TOE.

5.3.6 Observations and recommendations

[R34] The evaluated user guidance documentation shall contain all information required to use the TOE in a secure way as defined in the platform security target including recommendations on how to avoid residual vulnerabilities and unexpected behaviour.

[R35] However, in specific cases detailed information might be required in addition to the guidance documents such as:

- Observations on the evaluation results (e.g. specific TOE configuration for the evaluation),
- Recommendations/stipulations for the **Composite Product Evaluator**: specific information on use of the evaluation results (e.g. about specific testing necessary during a composition evaluation).

Any such observation or recommendation/stipulation may come from both the **Platform Evaluator** or the **Platform Certification Body**.

[8] Evaluation Methodology; depends on the version of CC chosen

6 Evaluation/Certification reports and Platform certificate validity

[R36] Results of a composite evaluation shall be provided to the **Composite Product Certification Body** in form of an Evaluation Technical Report for the composite product. This Composite Product ETR shall contain, amongst others, the final overall verdict for the composite evaluation based on the partial verdicts for each assurance component being in scope of the current composite evaluation. There shall be a reference to this CC supporting document in the Composite Product ETR and the Composite Product Certification Report.

[R37] As the composite product certificate covers also the platform, the composite product certificate validity is linked to the validity of the platform certificate.

[R38] The **Composite Product Certification Body** needs an up-to-date certificate or an assessment from the **Platform Certification Body** on the status of the platform certificate in question.

[R39] As a general rule the **Composite Product Certification Body** will ask for a reassessment of the platform if the date of the platform's ETR for Composition is more than one and a half year before the submission of the report containing the full results of the composition penetration tests. This reassessment consists of either a re-evaluation of the platform focussing on a renewal of the vulnerability analysis (surveillance task) or alternatively, a confirmation statement of the **Platform Certification Body** may be requested.

[R40] Note that in the case the entire composite product is set up as a chain of composite products constructed on top of each other (e.g. the platform itself is already a composite product) the maximum validity period of 18 months is related to the eldest ETR for Composition used in this chain of composite products. In addition, dependencies from a lower level ETR for Composition to a higher level ETR for Composition need to be considered when reusing the results in the composite evaluation on top.

[R41] Note also that if the platform's ETR for Composition was issued less than a year and a half ago before submission of the related composite evaluation tasks, but there was a major change in the state of the art in performing relevant attacks on the platform (e.g. a major change in the "Application of Attack Potential to Smart Cards" document [JIL AP] or a major change in attack methods or attack ratings) then the **Composite Product Certification Body** has the right to require a reassessment focusing on the new attack method.

[R42] Validity and relevance of the platform certificate for the current composite product certification shall be acknowledged by the **Composite Product Certification Body** and includes the determination of equivalence of single assurance components (and, hence, of assurance levels) belonging to different CC versions, if the platform certification was according to another CC version

than the current composite certification is. Such equivalence shall be established / acknowledged by the **Composite Product Certification Body** (see section 6.1).

[R43] The **Composite Product Certification Body** can issue a security certificate for the composite product, if

- the verdicts for the Composite Product ETR is PASS and
- validity and topicality of the platform certificate for the current composite product is acknowledged by the **Composite Product Certification Body**.

[R44] Note that, if the **Composite Product Evaluator** detects some failures resulting from Platform 'Open Samples' testing, the results are communicated to the **Composite Product Certification Body**. The **Composite Product Certification Body** shall then take appropriate steps together with the **Platform Certification Body**.

6.1 Composite evaluation based on different versions of CC

[R45] Due to CCRA rules, any new evaluation shall be performed using CC version 3.1, but today, a number of certified PPs are only available in CC version 2.3. Also a significant number of HW platforms are on the market that were only certified under CC version 2.3. Therefore, in accordance with discussions at the JIWG and CCDB level, the CBs allow to perform:

- evaluations under CC v3.1, with the security target conformant to PPs certified under CC version 2.3.
- a composite evaluation based on CCv3.1 using a platform that was certified under CC2.3.

[R46] The certification body allows this specific mixture of Common Criteria version based on the following rule:

- The assurance gained under a specific EAL is equivalent between version 2.3 and 3.1 of the CC.

[R47] Based on this rule it is also valid to state that vulnerability assurance family AVA_VLA.3 is equivalent to AVA_VAN.4 and AVA_VLA.4 is equivalent to AVA_VAN.5.

[R48] As a consequence there are no issues for composite evaluation except for the following augmentation:

ADV_IMP:

[R49] The PPs in CC v2.3 generally have the augmentation IMP.2, meaning that the security target in CC v3.1 should claim this augmentation to remain conformant to the PP. Considering that:

- IMP.2 in CC v3.1 has a dependency on ALC_CMC.5 which means more work than before; and

- the CC v3.1 Eurosmart PP (PP0035) does not claim the augmentation IMP.2, meaning that the composite product cannot claim this augmentation if the underlying IC is certified with CC v3.1.

[R50] The CBs will:

- consider the package EAL4 + DVS.2, IMP.2, VLA.4 in CC v2.3 equivalent to EAL4+ DVS.2, VAN.5 in CC v3.1.

- accept a security target in CC v3.1 with only IMP.1, but claiming a conformity to the PP v2.3 (i.e. despite the lack of IMP.2).

7 References

7.1 CCV2.3 documents

[CC23] CCMB-2005-08-001 : Part 1: Introduction and General Model, Version 2.3

CCMB-2005-08-002 : Part 2: Security Functional Requirements, Version 2.3

CCMB-2005-08-003 : Part 3: Security Assurance Requirements, Version 2.3

[CEM23] CCMB-2005-08-004 : Evaluation Methodology, Version 2.3

7.2 CC V3 documents

[CC3] CCMB-2009-07-001 : Part 1 Introduction and general model, Version 3.1, Revision 3

CCMB-2009-07-002 : Part 2 Security Functional components, Version 3.1, Revision 3

CCMB-2009-07-003 : Part 3 Security Assurance Components, Version 3.1, Revision 3

[CEM3] CCMB-2009-07-004 : Evaluation Methodology, Version 3.1, Revision 1

7.3 Supporting documents

[JIL AP] Joint Interpretation Library Application of Attack Potential to Smart Cards V2.7 February 2009

[CC AC] CCIMB-2004-02-009 Assurance continuity: CCRA requirements

Appendix 1: Composite-specific requirements

In the following, the Composite-specific developer and evaluator action elements as well as the evaluator actions (*work units*) belonging to the composition activities (cf. chapter 4 above) are defined. They require the evidences as listed in section 4.7.
These refinements to the assurance requirements aim to give the **Composite Product Developer** and **Evaluator** a precise guidance on which relevant aspects have to be described and assessed in the context of a composite evaluation and the tasks to be performed.
It allows the **Composite Product Certification Body** to check using the composite product ETR that the required (mandatory) tasks have properly been performed.
All composite-specific evaluator actions have to be documented according to the scheme rules and finalised by one of the verdicts PASS, FAIL or INCONCLUSIVE. As these actions are refinements of the traditional actions focused on the composition activities, these verdicts have to be integrated to the overall verdict.
This approach can be applied independent of the aimed evaluation assurance level (EAL) for the composite product. Where some evaluation activities are not applicable due to the EAL chosen, the related composite-specific tasks are also not expected to be applied.
The composite evaluation methodology described by the current document is in principle independent of the CC version (V2.x or V3). However, due to differences in the content details of the related assurance classes and for the sake of a simpler applicability, the composite-specific tasks are defined twice:

– Composite-specific tasks for a composite evaluation in CC **V2.x** in Appendix 1.1, and

– Composite-specific tasks for a composite evaluation in CC **V3** in Appendix 1.2.

So, if the CC version chosen for the composite evaluation is e.g. V2.3, the user of the current document shall directly address the description in Appendix 1.1; if the CC version chosen for the composite evaluation is e.g. V3, the user of the current document shall directly address the description in Appendix 1.2.
For convenience of composite-specific activities and associated work units identification, each refinement is named as *_COMP, where * is the name of the assurance class it is related to.

Appendix 1.1: Composite-specific tasks for a composite evaluation in CC V2.x

1. Consistency of composite product Security Target (ASE_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the ASE class listed in the following table. The other activities of ASE class do not require composite-specific work units.

Assurance family	Evaluation activity	Evaluation work unit	Composite-specific work unit
ASE_ENV	ASE_ENV.1.2E	ASE_ENV.1-4	ASE_COMP.1-6
		ASE_ENV.1-4	ASE_COMP.1-7
		ASE_ENV.1-4	ASE_COMP.1-8
		ASE_ENV.1-4	ASE_COMP.1-9
ASE_OBJ	ASE_OBJ.1.2E	ASE_OBJ.1-7	ASE_COMP.1-5
	ASE_OBJ.1.3C	ASE_OBJ.1-3	ASE_COMP.1-10
ASE_REQ	ASE_REQ.1.4C	ASE_REQ.1-8	ASE_COMP.1-3
	ASE_REQ.1.2E	ASE_REQ.1-24	ASE_COMP.1-4
		ASE_REQ.1-24	ASE_COMP.1-11
ASE_TSS	ASE_TSS.1.6C	ASE_TSS.1-7	ASE_COMP.1-1
		ASE_TSS.1-7	ASE_COMP.1-2

ASE_COMP.1 Consistency of Security Target

Objectives

1 The aim of this activity is to determine whether the Security Target of the composite product[9] does not contradict the Security Target of the underlying platform[10].

Application notes

2 These application notes aid the developer to create as well as the evaluator to analyze a composite Security Target and describe a general methodology for it. For detailed information / guidance please refer to the single work units below.

3 In order to create a composite Security Target the developer should perform the following steps:

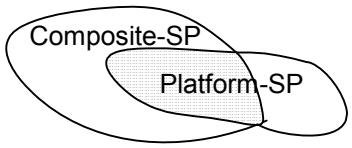
4 Step 1: The developer formulates a preliminary Security Target for the composite product (Composite-ST) using the standard code of practice.

[9] denoted by Composite-ST in the following

[10] denoted by Platform-ST in the following. Generally, a Security Target expresses a security policy for the TOE defined.

The Composite-ST can be formulated independently of the Security Target of the underlying platform (Platform-ST) – at least as long as there are no formal PP conformance claims.

5 Step 2: The developer determines the intersection of the Composite-ST and the Platform-ST analysing and comparing their TSF[11]:



6 Step 3: The developer determines under which conditions he can trust in and rely on the Platform-TSF being used by the Composite-ST without a new examination.

7 Having undertaken these steps the developer completes the preliminary Security Target for the composite product.

8 It is not mandatory that the platform is and the composite TOE is being certified according to same version of the CC. It is due to the fact that the application can rely on some security services of the platform, if (i) the assurance level of the platform covers the intended assurance level of the composite TOE and (ii) the platform's security certificate is valid and up-to-date. Equivalence of single assurance components (and, hence, of assurance levels) belonging to different CC versions shall be established / acknowledged by the Composite Product Certification Body, cf. chapter 6.

9 If a PP conformance is claimed (e.g. composite ST claim conformance to a PP that claims conformance to a hardware PP), the consistency check can be reduced to the elements of the Security Targets having not already been covered by these Protection Profiles.

The fact of compliance to a PP is not sufficient to avoid inconsistencies.
Assume the following situation, where → stands for "complies with"
Composite-ST → SW PP → HW PP ← Platform-ST
The SW PP may require any kind of conformance[12], but this does not change the 'additional elements' that the Platform-ST may introduce to the HW PP. In conclusion, these additions are not necessarily consistent with the composite-ST/SW PP additions: There is no scenario that ensures the consistency 'by construction'.
Note that consistency may not be direct matching: e.g. objectives for the platform environment may become objectives for the composite TOE.

[11] Because TSF enforce the Security Target (together with the TOE assurance measures). Please note that TSF means 'TOE Security Functions' in CC V2.x

[12] e.g. "exact", "strict" or "demonstrable" according to CC V2.4; there can only be "strict" for CC V2.1 to V2.3.

Dependencies:

10 No dependencies.

Developer action elements:

ASE_COMP.1.1D

11 The developer shall provide a statement of compatibility between the Composite Security Target and the Platform Security Target. This statement can be provided within the Composite Product Security Target.

Content and presentation of evidence elements:

ASE_COMP.1.1C

12 The statement of compatibility shall describe the separation of the Platform-TSF into relevant Platform-TSF being used by the Composite-ST and others.

ASE_COMP.1.2C

13 The statement of compatibility between the Composite Security Target and the Platform Security Target shall show (e.g. in form of a mapping) that the Security Targets of the composite product and of the underlying platform match, i.e. that there is no conflict between security environments, security objectives, and security requirements of the Composite Security Target and the Platform Security Target. It can be provided by indicating of the concerned elements directly in the Security Target for the composite product followed by explanatory text, if necessary.

Evaluator action elements:

ASE_COMP.1.1E

14 The evaluator shall confirm that information provided meets all requirements for content and presentation of evidence.

Evaluator actions:

Action ASE_COMP.1.1E

ASE_COMP.1.1C

ASE_COMP.1-1 The evaluator shall check that the statement of compatibility describes the separation of the Platform-TSF into relevant Platform-TSF being used by the Composite-ST and others.

15 This work unit relates to the *Step 2* of the *Application Notes* above. In order to determine the intersection area the evaluator considers the list of the Platform-TSF (given in the ST of the underlying platform) as its security services. To give an example, let us assume that there are the following Platform-TSF: Cryptographic functions RSA, AES, TDES, TRNG as well as tamper-resistance.

16 These Platform-TSF shall be separated in two groups:

– **IP_SF**: Irrelevant Platform-TSF not being used by the Composite-ST, and
– **RP_SF**: Relevant Platform-TSF being used by the Composite-ST.

17 The second group *RP_SF* exactly represents the intersection area in question. For example, *IP_SF* = {AES} and *RP_SF* = {RSA, TDES, TRNG, tamper-resistance}, i.e. AES is not used by the composite TOE, but all other Platform-TSF are used.

18 The amount of the intersection area (i.e. the content of the group *RP_SF*) results from the concrete properties of the Platform-ST and the Composite-ST. If the Composite-ST does not use any property of the Platform-ST and, hence, the intersection area is an empty set (*RP_SF* = {∅}), no further composite evaluation activities are necessary at all: In such a case there is a technical, but not a security composition.

19 The result of this work unit shall be integrated to the result of ASE_TSS.1.6C/ ASE_TSS.1-7.

ASE_COMP.1-2 The evaluator ***shall examine*** the statement of compatibility to determine that the list of the Platform-TSF being used by the Composite-ST is complete and consistent for the current composite TOE.

20 In order to determine the completeness of the list of the Platform-TSF being used by the Composite-ST, the evaluator shall verify that:

– Platform-TSF = IP_SF ∪ RP_SF
– Elements that belong to RP_SF actually reflect the composite TOE

21 In order to determine the consistency of the list of the Platform-TSF being used by the Composite-ST, the evaluator shall verify that there are no ambiguities and contradictory statements.

22 More details on the consistency analysis can be found in common CC documents.

23 The result of this work unit shall be integrated to the result of ASE_TSS.1.6C/ ASE_TSS.1-7.

ASE_COMP.1.2C

ASE_COMP.1-3 The evaluator ***shall check*** that the assurance requirements of the composite evaluation represent a subset of the assurance requirements of the underlying platform[13].

24 This work unit relates to the *Step 2* of the *Application Notes* above. In order to ensure a sufficient degree of trustworthiness of the Platform-TSF the evaluator compares the TOE assurance requirements[14] of the composite evaluation with those of the underlying platform. The evaluator decides that the degree of trustworthiness of the Platform-TSF

[13] Please note that assurance measures can be derived from assurance requirements in a direct way, e.g. as a one to one assignment.
[14] denoted by SAR in the following

is sufficient, if the Composite-SAR represent a subset of the Platform-SAR:

$$\text{Platform-SAR} \supseteq \text{Composite-SAR},$$

e.g. the EAL chosen for the composite evaluation does not exceed the EAL applied to the evaluation of the platform.

25 The result of this work unit shall be integrated to the result of ASE_REQ.1.4C/ ASE_REQ.1-8.

ASE_COMP.1-4 The evaluator ***shall examine*** the statement of compatibility to determine that all performed operations on the relevant TOE security functional requirements of the platform are appropriate for the Composite-ST.

26 This work unit relates to *Step 3* of the *Application Notes* above. The *relevant* TOE security functional requirements[15] of the platform are the functional requirements being enforced by the Platform-TSF of the group *RP_SF* (cf. the work unit ASE_COMP.1-1), or, shortly, being mapped (in the rationale of the Platform-ST) to the TSF belonging to the group *RP_SF*.

27 In order to perform this work unit the evaluator compares single parameters the *relevant* TOE security functional requirements of the platform with those of the composite evaluation. For example, the evaluator compares the properties of the component FCS_COP.1/RSA and determines that the Composite-ST requires a key length of 2048 bit and the Platform-ST enforces the RSA-function with a key length of 1024 and 2048 bit, i.e. this parameter of the platform is appropriate for the Composite-ST. Note, that the Composite-TSFR need not necessarily be the same as the Platform-TSFR, e.g. a trusted channel (FTP_ITC.1) in the composite product can be built using an RSA implementation (FCS_COP.1/RSA) of the platform.

28 The result of this work unit shall be integrated to the result of ASE_REQ.1.2E/ ASE_REQ.1-24.

ASE_COMP.1-5 The evaluator ***shall examine*** the statement of compatibility to determine that the relevant TOE security objectives of the Platform-ST are not contradictory to those of the Composite-ST.

29 This work unit relates to *Step 3* of the *Application Notes* above. The *relevant* TOE security objectives of the Platform-ST are those that are mapped to the *relevant* TOE security functional requirements of the Platform-ST (cf. the work unit ASE_COMP.1-4).

[15] TSFR

30 In order to perform this work unit the evaluator compares the *relevant* TOE security objectives of the Platform-ST with those of the Composite-ST and determines whether they are not contradictory.

31 The result of this work unit shall be integrated to the result of ASE_OBJ.1.2E/ ASE_OBJ.1-7.

ASE_COMP.1-6 The evaluator ***shall examine*** the statement of compatibility to determine that the relevant threats of the Platform-ST are not contradictory to those of the Composite-ST.

32 This work unit relates to *Step 3* of the *Application Notes* above. The evaluator compares the *relevant* threats (i.e. being mapped to the relevant TOE security objectives, cf. the work unit ASE_COMP.1-5) of the Platform-ST with those of the Composite-ST and determines whether they are not contradictory. The evaluator can decide on non-contradiction, if the threats of the Composite-ST referring to the platform-part of the composite product are covered by the threats of the Platform-ST. For example, there may be a threat T.Physical_Attack of the Composite-ST covered by the threat T.Tamper of the Platform-ST.

33 The result of this work unit shall be integrated to the result of ASE_ENV.1.2E/ ASE_ENV.1-4.

ASE_COMP.1-7 The evaluator ***shall examine*** the statement of compatibility to determine that the relevant organisational security policies of the Platform-ST are not contradictory to those of the Composite-ST.

34 This work unit relates to *Step 3* of the *Application Notes* above. The evaluator compares the *relevant* organisational security policies (i.e. being mapped to the relevant TOE security objectives, cf. the work unit ASE_COMP.1-5) of the Platform-ST with those of the Composite-ST and determines whether they are not contradictory.

35 Beyond it, a special organisational security policy OSP.Composite could be formulated within the Composite-ST, e.g. ‘The application is running on a certified platform and compatible with it’. Then the developer can define the special security objectives for the TOE and its environment exactly reflecting the conditions and restrictions of the certification report of the underlying platform.

36 The result of this work unit shall be integrated to the result of ASE_ENV.1.2E/ ASE_ENV.1-4.

ASE_COMP.1-8 The evaluator ***shall examine*** the statement of compatibility to determine that the relevant organisational security policies of the Platform-ST are not contradictory to the threats of the Composite-ST and vice versa.

37 This work unit relates to *Step 3* of the *Application Notes* above. The evaluator compares the *relevant* organisational security policies (i.e. being mapped to the relevant TOE security objectives, cf. the work unit ASE_COMP.1-5) of the Platform-ST with the threats of the Composite-ST and determines whether they are not contradictory.

38 An example for contradictive items: The organisational security policy of the Platform-ST "Cryptographic algorithms used shall be in accordance with international standards" is contradictory to the threat of the Composite-ST "An attacker discloses the secrets being used by the TOE proprietary cryptographic algorithm".

39 The result of this work unit shall be integrated to the result of ASE_ENV.1.2E/ ASE_ENV.1-4.

ASE_COMP.1-9 The evaluator ***shall examine*** the statement of compatibility to determine that the list of the assumptions of the Platform-ST being significant for the Composite-ST is complete and consistent for the current composite TOE.

40 This work unit relates to *Step 3* of the *Application Notes* above. In order to determine which assumptions of the Platform-ST are *significant* for the Composite-ST the evaluator analyses the assumptions of the Platform-ST and their separation in the following groups:

– **IrPA**: The assumptions being not relevant for the Composite-ST, e.g. the assumptions about the developing and manufacturing phases of the platform.
– **CfPA**: The assumptions being fulfilled by the Composite-ST *automatically*. Such assumptions of the Platform-ST can always be assigned to the TOE security objectives of the Composite-ST. Due to this fact they will be fulfilled either by the Composite-TSF or by the Composite-TAM automatically. To give an example, let there be an assumption A.Resp-Appl of the Platform-ST: 'All User Data are owned by Smartcard Embedded Software. Therefore, it must be assumed that security relevant User Data (especially cryptographic keys) are treated by the Smartcard Embedded Software as defined for the specific application context' and a TOE security objective OT.Key_Secrecy of the Composite-ST: 'The secrecy of the signature private key used for signature generation is reasonably assured against attacks with a high attack potential.' If the private key is the only sensitive data element, then the assumption A.Resp-Appl is covered by the TOE security objective OT.Key_Secrecy automatically.
– **SgPA**: The remaining assumptions of the Platform-ST belonging neither to the group *IrPA* nor *CfPA*. **Exactly this group makes up the *significant* assumptions for the Composite-ST**, which shall be included into the Composite-ST.

41 The result of this work unit shall be integrated to the result of ASE_ENV.1.2E/ ASE_ENV.1-4.

ASE_COMP.1-10 The evaluator ***shall examine*** the statement of compatibility to determine that the significant security objectives for the operational environments of the Platform-ST are not contradictory to those of the Composite-ST.

42 This work unit relates to *Step 3* of the *Application Notes* above. The *significant* security objectives for the operational environment of the Platform-ST are the security objectives for the operational environment being assigned to the assumptions classified as the group *SgPA* of the Platform-ST (cf. the work unit ASE_COMP.1-9).

43 In order to accomplish this work unit the evaluator compares the *significant* security objectives for the operational environment of the Platform-ST with those of the Composite-ST and determines whether they are not contradictory. If necessary, the *significant* security objectives for the operational environment of the Platform-ST shall be included into the Composite-ST and assigned to the assumptions from the group *SgPA*, cf. the work unit ASE_COMP.1-9. The inclusion is not necessary, if the Composite-ST already contains equivalent (or similar) security objectives (covering all relevant aspects).

44 Since assurance of the development and manufacturing environment of the platform is confirmed by the platform certificate, the respective platform-objectives, if any, belong to the group *IrPA*.

45 Assurance of development and manufacturing environment is usually completely addressed by the assurance class ALC, and, hence, requires no explicit security objective.

46 The result of this work unit shall be integrated to the result of ASE_OBJ.1.3C/ ASE_OBJ.1-3.

ASE_COMP.1-11 The evaluator ***shall examine*** the statement of compatibility to determine that the significant security functional requirements for the environment of the Platform-ST are not contradictory to those of the Composite-ST.

47 This work unit relates to *Step 3* of the *Application Notes* above. The evaluator compares the *significant* security functional requirements for the environment of the Platform-ST (i.e. being mapped to the *significant* security objectives for the environment, cf. the work unit ASE_COMP.1-10) with those of the Composite-ST and determines whether they are not contradictory. If necessary, the *significant* security functional requirements for the environment of the Platform-ST shall be included into the Composite-ST and assigned to the significant security objectives (cf. the work unit ASE_COMP.1-9). The inclusion is not necessary, if the Composite-ST already contains equivalent (or similar) security functional requirements (covering all relevant aspects). Note that non-IT requirements are optional.

48 The result of this work unit shall be integrated to the result of ASE_REQ.1.2E/ ASE_REQ.1-24.

2. Integration of composition parts (ACM_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the ACM class listed in the following table. The other activities of ACM class do not require composite-specific work units.

CC Assurance family	Evaluation activity	Evaluation work unit	Composite-specific work unit
ACM_CAP	ACM_CAP.2.5C	ACM_CAP.2-6	ACM_COMP.1-1

NB: If the level of the assurance requirement chosen is higher than those identified in this table, the composite-specific work unit is also applicable.

ACM_COMP.1 Integration of the application into the underlying platform

Objectives

49 The aim of this activity is to determine whether the correct version of the application is installed onto/into the correct version of the underlying platform.

Dependencies:

50 No dependencies.

Developer action elements:

ACM_COMP.1.1D

51 The developer shall provide components identification evidence; cf. item #7-(i) in Table 1, section 4.7.

Content and presentation of evidence elements:

ACM_COMP.1.1C

52 The components identification evidence shall show that the evaluated version of the application has been installed onto / embedded into the certified version of the underlying platform.

Evaluator action elements:

ACM_COMP.1.1E

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

Evaluator actions:

Action ACM_COMP.1.1E

ACM_COMP.1-1 The evaluator ***shall check*** the evidence that the evaluated version of the application has been installed onto / embedded into the correct, certified version of the underlying platform.

54 The *general* information of the CM capabilities is represented and has to be examined in the context of the assurance family ACM_CAP. The

special composite evaluator activity is to check the evidence of the version correctness for both parts of the composite product.

55 For the underlying platform, the evaluator shall determine that the actual identification of the platform is commensurate with the respective data in the platform certificate.

56 For the application, the relevant task is trivial due to the fact that the **Composite Product Evaluator** has to perform this task in the context of the assurance family ACM_CAP.

57 Components identification evidence can be supplied in two different ways: *technical* and *organisational*. A technical evidence of version correctness is being generated by the composite product itself: the platform and the application return – in each case – strings containing unambiguous version numbers as answers to the respective commands. E.g. it can be the return string of a command or the hard copy of the Windows-Information (like 'About'); in case of smart cards it can be an appropriate ATR.

58 A technical evidence of version correctness for hardware can also be supplied, if applicable, by reading off the unambiguous inscription on its surface. Note that there are no physical indication existing on most smart cards microcontrollers.

59 Technical evidence is recommended to be provided.

60 An organisational evidence of version correctness is being generated by the **Composite Product Integrator** on the basis of his configuration lists containing unambiguous version information of the platform and the application having been composed into the final composite product.

61 For example, in case of smart cards it can be an acknowledgement statement (e.g. configuration list) of the integrated circuit[16] manufacturer to the embedded software[17] manufacturer containing the evidence for the versions of the chip, the embedded software and its pre-personalisation parameters[18].

62 Organisational evidence is always possible and, hence, shall be provided.

63 The result of this work unit shall be integrated to the result of ACM_CAP.2.5C/ ACM_CAP.2-6 (or the equivalent higher components if a higher assurance level is selected).

[16] -> underlying platform

[17] -> application

[18] Any data supplied by the embedded software manufacturer that is injected into the non-volatile memory by the integrated circuits manufacturer. These data are for instance used for traceability and/or to secure shipment between phases (cf. [Smartcard IC Platform Protection Profile, Version 1.0, July 2001, registration number BSI-PP-0002], sec. 8.7).

3. Consistency of delivery procedures (ADO_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the ADO class listed in the following table. The other activities of ADO class do not require composite-specific work units.

Assurance family	Evaluation activity	Evaluation work unit	Composite-specific work units
ADO_DEL	ADO_DEL.1.1C	ADO_DEL.1-1	ADO_COMP.1-1
ADO_IGS	ADO_IGS.1.2E	ADO_IGS.1-2	ADO_COMP.1-2

NB: If the level of the assurance requirement chosen is higher than those identified in this table, the composite-specific work unit is also applicable.

ADO_COMP.1 Consistency check for delivery and acceptance procedures

Objectives

64 The aim of this activity is to determine whether the delivery procedures of **Platform** and **Application Developers** are compatible with the acceptance procedure of the **Composite Product Integrator**.

Dependencies:

65 No dependencies.

Developer action elements:

ADO_COMP.1.1D

66 The developer shall provide an evidence for delivery and acceptance compatibility; cf. item #8 in Table 1, section 4.7.

ADO_COMP.1.2D

67 The developer shall provide a configuration parameters evidence; cf. item #7-(ii) in Table 1, section 4.7.

Content and presentation of evidence elements:

ADO_COMP.1.1C

68 The evidence for delivery and acceptance compatibility shall show that the delivery procedures of the Platform and Application Developers are compatible with the acceptance procedure of the Composite Product Integrator.

ADO_COMP.1.2C

69 The configuration parameters evidence shall show that configuration parameters prescribed by the Platform and Application Developers are actually being used by the Composite Product Integrator.

Evaluator action elements:

ADO_COMP.1.1E

70 The evaluator shall confirm that the evidence for delivery compatibility is complete, coherent, and internally consistent.

ADO_COMP.1.2E

71 The evaluator shall confirm that the information provided meets all requirements for presentation of evidence.

Evaluator actions:

Action ADO_COMP.1.1E

ADO_COMP.1-1 The evaluator ***shall examine*** the evidence for compatibility of delivery interfaces to determine that delivery procedures of the Platform and Application Developers are compatible with the acceptance procedure of the Composite Product Integrator.

72 The *general* information of the delivery procedures is represented and has to be examined in the context of the assurance family ADO_DEL. The *additional* composite activity of the evaluator is to examine each delivery interface between the **Platform Developer** and the **Composite Product Integrator** on the one side and between the **Application Developer** and the **Composite Product Integrator** on the other side. As a result, the evaluator confirms or disproves the justification for delivery compatibility.

73 If there are no delivery interfaces between the **Platform** and **Application Developers** and the **Composite Product Integrator** or the assurance package chosen does not contain the family ADO_DEL (e.g. EAL1), this work unit is not applicable.

74 The result of this work unit shall be integrated to the result of ADO_DEL.1.1C/ ADO_DEL.1-1 (or the equivalent higher components if a higher assurance level is selected).

Action ADO_COMP.1.2E

ADO_COMP.1-2 The evaluator ***shall examine*** the evidence for using configuration parameters to determine that the Composite Product Integrator uses the configuration parameters prescribed by the Platform and Application Developers.

75 The *general* information of the configuration parameters required is represented and has to be examined in the context of the assurance family ADO_IGS. The *special* evaluator activity is to examine the developer's evidence and to decide whether the **Composite Product Integrator** appropriately treats this *special subset* of the configuration parameters.

76 For example, for a Java Card as composite TOE, the Card Issuer has to set all parameters as prescribed by the Java Card Platform and the Applet Developers while installing the applet onto the Java Card platform; cf. Table 3, section 4.7.

77 The result of this work unit shall be integrated to the result of ADO_IGS.1.2E/ ADO_IGS.1-2.

4. Composite design compliance (ADV_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the ADV class listed in the following table. The other activities of ADV class do not require composite-specific work units.

CC Assurance family	Evaluation activity	Evaluation work unit	Composite specific work unit
ADV_HLD	ADV_HLD.1.2E	ADV_HLD.1-9	ADV_COMP.1-1
ADV_IMP	ADV_IMP.1.2E	ADV_IMP.1−4	ADV_COMP.1-1
ADV_INT	ADV_INT.1.2E	ADV_INT.1-4	ADV_COMP.1-2
ADV_LLD	ADV_LLD.1.2E	ADV_LLD.1−11	ADV_COMP.1-1

NB: If the level of the assurance requirement chosen is higher than those identified in this table, the composite-specific work unit is also applicable.

ADV_COMP.1 Design compliance with the platform certification report, guidance and ETR_COMP

Objectives

78 The aim of this activity is to determine whether the requirements on the application, imposed by the underlying platform, are fulfilled in the composite product.

Application notes

79 The requirements on the application, imposed by the underlying platform, can be formulated in the relevant certification report (e.g. in form of constraints and recommendations), user guidance and ETR_COMP (in form of observations and recommendations) for the platform. The developer of the composite product shall regard each of these sources, if available (cf. Table 2, section 4.7), and implement the composite product in such a way that the applicable requirements are fulfilled.

80 The TSF of the composite product are represented at various levels of abstraction in the families of the development class ADV. Experiential, the appropriate levels of design representation for examining, whether the requirements of the platform are fulfilled by the composite product, are the low-level design and the implementation. In case, these design representation levels are not available (e.g. due to the assurance package chosen), the high-level design can be used for providing the relevant information.

81 Due to the definition of the composite TOE (cf. section 2.1 ‘Definitions’) the interface between the underlying platform and the application is the *internal* one, hence, a functional specification (ADV_FSP) as representation level is not appropriate for analysing the design compliance.

82 Since consistency of the composite product security policy has already been considered in the context of the Security Target in the assurance family ASE_COMP (see page 29 above), there is no necessity to consider non-contradictoriness of the security policy model (ADV_SPM) of the composite TOE and the security policy model of the underlying platform.

83 There is no affinity between the family correspondence demonstration (ADV_RCR) and the examination of the design compliance.

Dependencies:

84 No dependencies.

Developer action elements:

ADV_COMP.1.1D

85 The developer shall provide a design compliance justification; cf. item #6 as well as items #3, #4, #5 in Table 1, section 4.7.

Content and presentation of evidence elements:

ADV_COMP.1.1C

86 The design compliance justification shall provide a rationale for design compliance – on an appropriate representation level – of how the requirements on the application, imposed by the underlying platform, are fulfilled in the composite product.

Evaluator action elements:

ADV_COMP.1.1E

87 The evaluator shall confirm that the rationale for design compliance is complete, coherent, and internally consistent.

Evaluator actions:

Action ADV_COMP.1.1E

ADV_COMP.1-1 The evaluator ***shall examine*** the rationale for design compliance to determine that all applicable requirements on the application, imposed by the underlying platform, are fulfilled by the composite product.

88 In order to perform this work unit the evaluator shall use the rationale for design compliance as well as the TSF representation on the ADV_LLD and ADV_IMP levels on the one side and the input of the platform developer in form of the certification report, guidance and ETR_COMP on the other side. The evaluator shall analyse which platform requirements are applicable for the current composite product. The evaluator shall compare each of the applicable requirements with the actual specification and/or implementation of the composite product and determine, for each requirement, whether it is fulfilled. As result, the evaluator confirms or disproves the rationale for design compliance.

89 For example, platform guidance may require the application to perform a special start-up sequence testing the current state of the platform and initialising its self-protection mechanisms. Such information might be

found in the description of low-level design ADV_LLD and/or ADO_IGS of the composite TOE.

90 The appropriate representation level (ADV_LLD and ADV_IMP), what the analysis is being performed on, can be chosen and mixed flexibly depending on the concrete composite TOE and the requirement in question. Where it is not self-explaining, the evaluator shall justify why the representation level chosen is appropriate. In case there is no information available on the representation levels ADV_LLD and ADV_IMP due to the assurance package chosen (e.g. EAL2, EAL3), the high-level design ADV_HLD shall be used for providing the relevant information and performing the corresponding analysis.

91 The evaluator activities in the context of this work unit can be spread over different single evaluation aspects (e.g. over ADV_LLD and ADV_IMP). In this case the evaluator performs the partial activity in the context of the corresponding single evaluation aspect. Then the notation for this work unit shall be ADV_COMP.1-1-HLD, ADV_COMP.1-1-LLD and ADV_COMP.1-1-IMP, respectively.

92 If the assurance package chosen does not contain the families ADV_HLD, ADV_LLD or ADV_IMP (e.g. EAL1), this work unit is not applicable (cf. *Application Note* above).

93 The result of this work unit shall be integrated to the result of ADV_HLD.1.2E/ ADV_HLD.1-9[19], ADV_LLD.1.2E/ ADV_LLD.1−11, ADV_IMP.1.2E/ ADV_IMP.1−4 (or the equivalent higher components if a higher assurance level is selected).

ADV_COMP.1-2 The evaluator ***shall check*** the TSF internals of the composite TOE to determine that they do not contradict any design requirement imposed by the underlying platform.

94 The TSF internals are represented and evaluated in the context of the assurance family ADV_INT. The evaluator shall compare the internal structure of the TSF with the design requirements of the platform and search for obvious contradictions.

95 If there are no requirements of the platform concerning the TSF internal structure or the assurance package chosen does not contain the family ADV_INT, this work unit is not applicable.

96 The result of this work unit shall be integrated to the result of ADV_INT.1.2E/ ADV_INT.1-4 (or the equivalent higher components if a higher assurance level is selected).

[19] ADV_HLD.3.2E might be not relevant for this activity due to the fact, that the evaluator should use ADV_LLD, if available.

5. Composite functional testing (ATE_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the ATE class listed in the following table. The other activities of ATE class do not require composite-specific work units.

CC Assurance family	Evaluation activity	Evaluation work unit	Composite specific work unit
ATE_COV	ATE_COV.1.1C	ATE_COV.1-1	ATE_COMP.1-1
			ATE_COMP.1-2
ATE_FUN	ATE_FUN.1.3C	ATE_FUN.1−6	ATE_COMP.1-1
ATE_IND	ATE_IND.1.2E	ATE_IND.1−5	ATE_COMP.1-3

NB: If the level of the assurance requirement chosen is higher than those identified in this table, the composite-specific work unit is also applicable.

ATE_COMP.1 Composite product functional testing

Objectives

97 The aim of this activity is to determine whether composite product *as a whole* exhibits the properties necessary to satisfy the functional requirements of its Security Target.

Application notes

98 A composite product can be tested *separately* and *integrative*. Separate testing means that the platform and the application are being tested independent of each other. A lot of tests of the platform may have been performed within the scope of its accomplished evaluation. The application may be tested on a simulator or an emulator, which represent a virtual machine.
Integration testing means that the composite product is being tested as it is: the application is running on the platform.

99 Some TSF can depend on properties of the underlying platform as well as of the application (e.g. correctness of the measures of the composite product to withstand a side channel attack or of the TSF implementing tamper resistance against physical attacks). In such a case the TSF shall be tested on the final composite product, but not on a simulator or an emulator.

100 This activity focuses exclusively on testing of the composite product *as a whole* and represents merely *partial efforts* within the general test approach being covered by the assurance ATE. These integration tests shall be specified and performed, whereby the approach of the standard[20] assurance families of the class ATE shall be applied.

[20] i.e. as required by CEM

101 - A correct behaviour of the Platform-TSF being relevant for the Composite-ST (the group *RP_SF* in the work unit ASE_COMP.1-1 above), and
- absence of exploitable vulnerabilities (sufficient effectiveness) in the context of the Platform-ST are confirmed by the valid Platform Certificate, cf. chapter 6 above.

Dependencies:

102 No dependencies.

Developer action elements:

ATE_COMP.1.1D

103 The developer shall provide a set of tests as required by the assurance package chosen.

ATE_COMP.1.2D

104 The developer shall provide the composite TOE for testing.

Content and presentation of evidence elements:

ATE_COMP.1.1C

105 Content and presentation of the specification and documentation of the *integration* tests shall correspond to the standard[21] requirements of the assurance families ATE_FUN and ATE_COV.

ATE_COMP.1.2C

106 The composite TOE provided shall be suitable for testing.

Evaluator action elements:

ATE_COMP.1.1E

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

ATE_COMP.1.2E

108 The evaluator shall specify, perform and document a set of own *integration* tests to confirm that the composite TOE operates as specified.

Evaluator actions:

Action ATE_COMP.1.1E

ATE_COMP.1-1 The evaluator ***shall examine*** that the developer performed the *integration* tests for all TSF having to be tested on the composite product as a whole.

109 In order to perform this work unit the evaluator shall analyse, for each TSF, whether it directly depends on security properties of the platform and of the application. Then the evaluator shall verify that the *integration* tests performed by the developer cover at least all such TSF.

110 If the assurance package chosen does not contain the families ATE_FUN and ATE_COV (e.g. EAL1), this work unit is not applicable.

[21] i.e. as defined by CEM

111 The result of this work unit shall be integrated to the result of ATE_COV.1.1C/ ATE_COV.1-1 and ATE_FUN.1.3C/ ATE_FUN.1−6 (or the equivalent higher components if a higher assurance level is selected).

Action ATE_COMP.1.2E

ATE_COMP.1-2 The evaluator ***shall determine*** the minimal amount of the *integration* tests being necessary for the current composite evaluation.

112 The evaluator shall perform the following steps:

1) The evaluator determines the share of the platform part of the TOE in enforcing of the Composite-ST. In order to do this, the evaluator refers to the TOE design (ADV_HLD might possess an appropriate representation level for this purpose) as well as to the Composite-ST and lists all Composite-SFRs using the security services of the platform[22]. Hereby the evaluator shall understand/document, for each such Composite-SFR, what concretely the platform does (so called *platform's share*).

For example, there is a Composite-SFR FCS_CKM.1 fulfilled by the TSF 'Key Generation'. The share of the platform in implementing this SFR is generating random numbers for key generation.

2) The evaluator checks, for each such Composite-SFR, whether the *platform's share* has been covered by the Platform Certificate; in order to do this the evaluator refers to the platform user guidance, ETR_COMP and the platform certification report.

For our example with the random number generator (RNG), the evaluator might find an advice in ETR_COMP that (i) the RNG is regularly being tested (online tests), (ii) its behaviour concerning power consumption analysis was evaluated and (iii) a sufficient (for generation of static keys) random numbers quality was confirmed by the Platform Certificate. In such a case the evaluator can decide that no *integration* tests are necessary for FCS_CKM.1/Key Generation[23].

The next example represents a situation, where additional *integration* tests are necessary. There let be a Composite-SFR FPT_EMSEC.1 (electromagnetic emanation) fulfilled by the following TSF:
- 'User Authentication'; the platform's share in implementing this SFR is scrambling the reference authentication data stored in the TOE;
- 'Key Generation'; the platform's share in implementing this SFR is disguising information about the value of the key being generated while

[22] Such security services of the platform are represented by the group *RP_SF, cf.* the work unit ASE_COMP.1-1 above

[23] Of course, the evaluator might (and, perhaps, would) decide to test 'Key Generation'-TSF as an integration test on the final TOE. A methodological reason for this test would then be rather a general check of the respective functionality, but not the confirmation of security behaviour of the platform-RNG.

observing power consumption;
- 'Signature Creation'; the platform's share in implementing this SFR is disguising information about the value of the signature key being used while observing power consumption;

For scrambling, the evaluator might find the advice in ETR_COMP that the platform scrambling engine has been evaluated, but it must be correctly initialised by the application. Hence, the evaluator has to specify an integration test for initialising the scrambling engine.
For disguising during key generation, the evaluator might find no advices in ETR_COMP. Hence, the evaluator has to assume that this behaviour of the platform was not included in consideration. Therefore, the evaluator has to specify an integration test for power consumption while generating the signature keys.
For disguising during signature creation, the evaluator might find an advice in ETR_COMP that DPA and SPA on RSA have been evaluated, but no advice of Timing on RSA. Hence, the evaluator has to assume that the Timing-on-RSA behaviour of the platform was not included in consideration. Therefore, the evaluator has to specify an integration Timing-on-RSA test for power consumption while using the signature key.

3) The evaluator refers to ETR_COMP and checks for any explicit requirements for performing tests in the context of the composite evaluation. Such explicit requirements might sound like 'Such test has to be performed during the SW evaluation'.

All platform tests being necessary for the current composite evaluation, but not covered by the Platform Certificate, and all tests explicitly required by ETR_COMP, encompass the minimal amount of the integration tests being necessary for the current composite evaluation.

This work unit is the complementary part to the work unit ASE_COMP.1-1: In ASE_COMP.1-1 the evaluator determines, on which part of the Platform-ST the Composite-ST can rely (the group *RP_SF*); in the current work unit the evaluator determines, whether the Composite-ST can also rely on the platform's *functional* behaviour being not covered by the Platform-ST.

113 The result of this work unit shall be integrated to the result of ATE_COV.1.1C/ATE_COV.1-1 (or the equivalent higher components if a higher assurance level is selected).

ATE_COMP.1-3 The evaluator ***shall perform*** the standard evaluator actions in the context of the assurance family ATE_IND on the set of the *integration* tests using the composite product as a whole.

114 The set of the *integration* tests for this activity shall embrace at least the minimal amount of the integration tests as determined in the previous work unit.

115 The result of this work unit shall be integrated to the result of ATE_IND.1.2E/ATE_IND.1−5 (or the equivalent higher components if a higher assurance level is selected).

6. Composite vulnerability assessment (AVA_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the AVA class listed in the following table. The other activities of AVA class do not require composite-specific work units.

CC Assurance family	Evaluation activity	Evaluation work unit	Composite-specific work unit
AVA_VLA	AVA_VLA.1.2E	AVA_VLA.1−4	AVA_COMP.1-1
	AVA_VLA.2.4E	AVA_VLA.2-11	AVA_COMP.1-2
	AVA_VLA.2.4E	AVA_VLA.2-12	AVA_COMP.1-2
	AVA_VLA.2.4E	AVA_VLA.2-13	AVA_COMP.1-2

NB: If the level of the assurance requirement chosen is higher than those identified in this table, the composite-specific work unit is also applicable.

AVA_COMP.1 Composite product vulnerability assessment

Objectives

116 The aim of this activity is to determine the exploitability of flaws or weaknesses in the composite TOE *as a whole* in the intended environment.

Application notes

117 This activity focuses exclusively on vulnerability assessment of the composite product *as a whole* and represents merely *partial efforts* within the general approach being covered by the standard[24] assurance families of the class AVA.

118 The results of the vulnerability assessment for the underlying platform represented in the ETR_COMP can be reused, if they are up to date and all composite activities for correctness – ASE_COMP.1, ACM_COMP.1, ADO_COMP.1, ADV_COMP.1 and ATE_COMP.1 – are finalised with the verdict PASS.

119 Due to composing of the platform and the application a new quality arises, which can cause additional vulnerabilities of the platform which might be not mentioned in the ETR_COMP.

[24] i.e. defined by CEM

Dependencies:

120 No dependencies.

Developer action elements:

AVA_COMP.1.1D

121 The developer shall provide the vulnerability assessment for the composite product, where aspects of interaction between the platform and the application are addressed.

AVA_COMP.1.2D

122 The developer shall provide the composite TOE for penetrating testing.

Content and presentation of evidence elements:

AVA_COMP.1.1C

123 Content and presentation of evidence elements shall correspond to the requirements of the assurance class AVA as defined by CEM. The focus of this Composite-special information lies on the aspects of interaction between the platform and the application.

AVA_COMP.1.2C

124 The composite TOE provided shall be suitable for testing *as a whole*.

Evaluator action elements:

AVA_COMP.1.1E

125 The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence. The focus of this Composite-special information lies on the aspects of interaction between the platform and the application.

AVA_COMP.1.2E

126 The evaluator shall conduct penetration testing of the composite product *as a whole* building on the developer vulnerability analysis, to ensure that obvious and identified vulnerabilities have been addressed.

Evaluator actions:

Action AVA_COMP.1.1E

AVA_COMP.1-1 The evaluator ***shall examine*** the results of the vulnerability assessment for the underlying platform to determine that they can be reused for the composite evaluation.

127 The results of the vulnerability assessment for the underlying platform are usually represented in the ETR_COMP. They can be reused, if they are up to date and all composite activities for correctness – ASE_COMP.1, ACM_COMP.1, ADO_COMP.1, ADV_COMP.1 and ATE_COMP.1 – are finalised with the verdict PASS. The evaluator shall also consider the relevant determinations in the Platform Certification Report. For validity of the platform security certificate please refer to chapter 6 above.

128 The result of this work unit shall be integrated to the result of AVA_VLA.1.2E/AVA_VLA.1−4 (or the equivalent higher components if a higher assurance level is selected).

Action AVA_COMP.1.2E

AVA_COMP.1-2 The evaluator ***shall specify, conduct and document*** penetration testing of the composite product *as a whole*, using the standard approach of the assurance family AVA_VLA.

129 If the correctness activities – ASE_COMP.1, ACM_COMP.1, ADO_COMP.1, ADV_COMP.1 and ATE_COMP.1 – are finalised with the verdict PASS and the certificate for the platform covers all security properties needed for the composite product, composing of the platform and the application must not create additional vulnerabilities of the platform.

130 If the evaluator determined that composing of the platform and the application creates additional vulnerabilities of the platform[25], a contradiction to the verdict PASS for the correctness activities (see paragraph 118 above) has to be supposed or the certificate for the platform does not cover all security properties needed for the current composite product.

131 If the assurance package chosen does not contain the family AVA_VLA (e.g. EAL1), this work unit is not applicable.

132 The result of this work unit shall be integrated to the result of AVA_VLA.2.4E/ AVA_VLA.2-11, AVA_VLA.2.4E/ AVA_VLA.2-12 and AVA_VLA.2.4E/ AVA_VLA.2-13 (or the equivalent higher components if a higher assurance level is selected).

[25] not mentioned in the ETR_COMP

Appendix 1.2: Composite-specific tasks for a composite evaluation in CC V3.1

1. Consistency of composite product Security Target (ASE_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the ASE class listed in the following table. The other activities of ASE class do not require composite-specific work units.

CC assurance family	Evaluation activity	Evaluation work unit	Composite-specific work unit
ASE_OBJ	ASE_OBJ.2.1C	ASE_OBJ.2-1	ASE_COMP.1-5
	ASE_OBJ.2.1C	ASE_OBJ.2-1	ASE_COMP.1-10
	ASE_OBJ.2.3C	ASE_OBJ.2-3	ASE_COMP.1-10
ASE_REQ	ASE_REQ.1.6C	ASE_REQ.1-10	ASE_COMP.1-1
	ASE_REQ.2.9C.	ASE_REQ.2-13	ASE_COMP.1-1
	ASE_REQ.1.6C	ASE_REQ.1-10	ASE_COMP.1-2
	ASE_REQ.2.9C.	ASE_REQ.2-13	ASE_COMP.1-2
	ASE_REQ.2.8C	ASE_REQ.2-12	ASE_COMP.1-3
	ASE_REQ.2.3C	ASE_REQ.2-4	ASE_COMP.1-4
ASE_SPD	ASE_SPD.1.1C	ASE_SPD.1-1	ASE_COMP.1-6
	ASE_SPD.1.3C	ASE_SPD.1-3	ASE_COMP.1-7
	ASE_SPD.1.3C	ASE_SPD.1-3	ASE_COMP.1-8
	ASE_SPD.1.4C	ASE_SPD.1-4	ASE_COMP.1-9

ASE_COMP.1 Consistency of Security Target

Objectives

1 The aim of this activity is to determine whether the Security Target of the composite product[26] does not contradict the Security Target of the underlying platform[27].

Application notes

2 These application notes aid the developer to create as well as the evaluator to analyze a composite Security Target and describe a general methodology for it. For detailed information / guidance please refer to the single work units below.

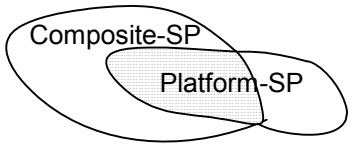
3 In order to create a composite Security Target the developer should perform the following steps:

[26] denoted by Composite-ST in the following

[27] denoted by Platform-ST in the following. Generally, a Security Target expresses a security policy for the TOE defined.

4 Step 1: The developer formulates a preliminary Security Target for the composite product (the Composite-ST) using the standard code of practice. The Composite-ST can be formulated independently of the Security Target of the underlying platform (Platform-ST) – at least as long as there are no formal PP conformance claims.

5 Step 2: The developer determines the intersection of the Composite-ST and the Platform-ST analysing and comparing their TOE Security Functionality (TSF) [28][29]:



6 Step 3: The developer determines under which conditions he can trust in and rely on the Platform-TSF being used by the Composite-ST without a new examination.

7 Having undertaken these steps the developer completes the preliminary Security Target for the composite product.

8 It is not mandatory that the platform is and the composite TOE is being certified according to same version of the CC. It is due to the fact that the application can rely on some security services of the platform, if (i) the assurance level of the platform covers the intended assurance level of the composite TOE and (ii) the platform's security certificate is valid and up-to-date. Equivalence of single assurance components (and, hence, of assurance levels) belonging to different CC versions shall be established / acknowledged by the Composite Product Certification Body, cf. chapter 6.

9 If a PP conformance is claimed (e.g. composite ST claim conformance to a PP that claims conformance to a hardware PP), the consistency check can be reduced to the elements of the Security Target having not already been covered by these Protection Profiles.

The fact of compliance to a PP is not sufficient to avoid inconsistencies.
Assume the following situation, where → stands for "complies with"
Composite-ST → SW PP → HW PP ← platform-ST
The SW PP may require any kind of conformance[30], but this does not change the 'additional elements' that the platform-ST may introduce to the HW PP. In conclusion, these additions are not necessarily consistent

[28] because the TSF enforce the Security Target (together with organisational measures enforcing security objectives for the operational environment of the TOE).
[29] The comparison shall be performed on the abstraction level of SFRs. If the developer defined security functionality groups (TSF-groups) in the TSS part of his Security Target, the evaluator should also consider them in order to get a better understanding for the context of the security services offered by the TOE.
[30] e.g. "strict" or "demonstrable" according to CC V3.

with the composite-ST/SW PP additions: There is no scenario that ensures the consistency 'by construction'.
Note that consistency may not be direct matching: e.g. objectives for the platform environment may become objectives for the composite TOE.

Dependencies:

10 No dependencies.

Developer action elements:

ASE_COMP.1.1D

11 The developer shall provide a statement of compatibility between the Composite Security Target and the Platform Security Target. This statement can be provided within the Composite Product Security Target.

Content and presentation of evidence elements:

ASE_COMP.1.1C

12 The statement of compatibility shall describe the separation of the Platform-TSF into relevant Platform-TSF being used by the Composite-ST and others.

ASE_COMP.1.2C

13 The statement of compatibility between the Composite Security Target and the Platform Security Target shall show (e.g. in form of a mapping) that the Security Targets of the composite product and of the underlying platform match, i.e. that there is no conflict between security environments, security objectives, and security requirements of the Composite Security Target and the Platform Security Target. It can be provided by indicating of the concerned elements directly in the Security Target for the composite product followed by explanatory text, if necessary.

Evaluator action elements:

ASE_COMP.1.1E

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

Evaluator actions:

Action ASE_COMP.1.1E
ASE_COMP.1.1C

ASE_COMP.1-1 The evaluator shall check that the statement of compatibility describes the separation of the Platform-TSF into relevant Platform-TSF being used by the Composite-ST and others.

15 Please note that TSF means 'TOE Security Functionality' in CC V3, whereby the TSF content is represented by SFRs[31]. The respective TOE summary specification (TSS) shall provide, for each SFR, a description on how each SFR is met[32]. The evaluator shall use this description in order to understand the contextual frame of the SFRs.

[31] security functional requirements
[32] cf. CC part 3, ASE_TSS.1.1C

If the developer defined security functionality groups (TSF-groups) in the TSS part of his Security Target as such contextual frame of the SFRs, the evaluator should also consider them in order to get a better understanding for the context of the security services offered by the TOE.

16 This work unit relates to the *Step 2* of the *Application Notes* above. In order to determine the intersection area the evaluator considers the list of the Platform-SFRs (given in the ST of the underlying platform) as single properties of the platform's security services.
To give an example, let us assume that there are the following Platform-SFRs: Cryptographic operations FCS_COP.1/RSA, FCS_COP.1/AES, FCS_COP.1/EC, FCS_COP.1/RNG as well as tamper-resistance FPT_PHP.3.

17 These Platform-SFRs shall be separated in two groups:

– **IP_SFR**: Irrelevant Platform-SFRs not being used by the Composite-ST, and
– **RP_SFR**: Relevant Platform-SFRs being used by the Composite-ST.

18 The second group *RP_SFR* exactly represents the intersection area in question. For example, *IP_SFR* = {FCS_COP.1/AES} and *RP_SFR* = {FCS_COP.1/RSA, FCS_COP.1/EC, FCS_COP.1/RNG, FPT_PHP.3}, i.e. AES is not used by the composite TOE, but all other Platform-SFRs are used.

19 The amount of the intersection area (i.e. the content of the group *RP_SFR*) results from the concrete properties of the Platform-ST and the Composite-ST. If the Composite-ST does not use any property of the Platform-ST and, hence, the intersection area is an empty set (*RP_SFR* = {∅}), no further composite evaluation activities are necessary at all: In such a case there is a technical, but not a security composition.

20 The result of this work unit shall be integrated to the result of ASE_REQ.1.6C/ ASE_REQ.1-10 (or the equivalent higher components if a higher assurance level is selected) and ASE_REQ.2.9C/ ASE_REQ.2-13.

ASE_COMP.1-2 The evaluator ***shall examine*** the statement of compatibility to determine that the Platform-TSF being used by the Composite-ST is complete and consistent for the current composite TOE.

21 In order to determine the completeness of the list of the Platform-TSF being used by the Composite-ST, the evaluator shall verify that:

– Platform-SFR = IP_SFR ∪ RP_SFR
– elements that belong to RP_SFR actually reflect the composite TOE

22 In order to determine the consistency of the list of the Platform-TSF being used by the Composite-ST, the evaluator shall verify that there are no ambiguities and contradictory statements.

23 More details on the consistency analysis can be found in common CC documents.

24 The result of this work unit shall be integrated to the result of ASE_REQ.1.6C/ ASE_REQ.1-10 (or the equivalent higher components if a higher assurance level is selected) and ASE_REQ.2.9C/ ASE_REQ.2-13.

ASE_COMP.1.2C

ASE_COMP.1-3 The evaluator ***shall check*** that the security assurance requirements of the composite evaluation represent a subset of the security assurance requirements of the underlying platform.

25 This work unit relates to the *Step 2* of the *Application Notes* above. In order to ensure a sufficient degree of trustworthiness of the Platform-TSF the evaluator compares the TOE security assurance requirements[33] of the composite evaluation with those of the underlying platform. The evaluator decides that the degree of trustworthiness of the Platform-TSF is sufficient, if the Composite-SAR represent a subset of the Platform-SAR:

$$\text{Platform-SAR} \supseteq \text{Composite-SAR},$$

e.g. the EAL chosen for the composite evaluation does not exceed the EAL applied to the evaluation of the platform.

26 The result of this work unit shall be integrated to the result of ASE_REQ.2.8C/ ASE_REQ.2-12.

ASE_COMP.1-4 The evaluator ***shall examine*** the statement of compatibility to determine that all performed operations on the relevant TOE security functional requirements of the platform are appropriate for the Composite-ST.

27 This work unit relates to *Step 3* of the *Application Notes* above. The *relevant* TOE security functional requirements of the platform are exactly the elements of the group *RP_SFR* (cf. the work unit ASE_COMP.1-1).

28 In order to perform this work unit the evaluator compares single parameters of the *relevant* SFRs of the platform with those of the composite evaluation. For example, the evaluator compares the properties of the respective components FCS_COP.1/RSA and determines that the Composite-ST requires a key length of 2048 bit and the Platform-ST enforces the RSA-function with a key length of 1024 and 2048 bit, i.e. this parameter of the platform is appropriate for the Composite-ST. Note, that the Composite-SFRs need not necessarily be the same as the Platform-SFRs, e.g. a trusted channel (FTP_ITC.1) in the

[33] denoted by SAR in the following

composite product can be built using an RSA implementation (FCS_COP.1/RSA) of the platform.

29 The result of this work unit shall be integrated to the result of ASE_REQ.2.3C/ ASE_REQ.2-4.

ASE_COMP.1-5 The evaluator ***shall examine*** the statement of compatibility to determine that the relevant TOE security objectives of the Platform-ST are not contradictory to those of the Composite-ST.

30 This work unit relates to *Step 3* of the *Application Notes* above. The *relevant* TOE security objectives of the Platform-ST are those that are mapped to the *relevant* SFRs of the Platform-ST (cf. the work unit ASE_COMP.1-4).

31 In order to perform this work unit the evaluator compares the *relevant* TOE security objectives of the Platform-ST with those of the Composite-ST and determines whether they are not contradictory.

32 The result of this work unit shall be integrated to the result of ASE_OBJ.2.1C/ ASE_OBJ.2-1.

ASE_COMP.1-6 The evaluator ***shall examine*** the statement of compatibility to determine that the relevant threats of the Platform-ST are not contradictory to those of the Composite-ST.

33 This work unit relates to *Step 3* of the *Application Notes* above. The evaluator compares the *relevant* threats (i.e. being mapped to the relevant TOE security objectives, cf. the work unit ASE_COMP.1-5) of the Platform-ST with those of the Composite-ST and determines whether they are not contradictory. The evaluator can decide on non-contradiction, if the threats of the Composite-ST referring to the platform-part of the composite product are covered by the threats of the Platform-ST. For example, there may be a threat T.Physical_Attack of the Composite-ST covered by the threat T.Tamper of the Platform-ST.

34 The result of this work unit shall be integrated to the result of ASE_SPD.1.1C/ ASE_SPD.1-1.

ASE_COMP.1-7 The evaluator ***shall examine*** the statement of compatibility to determine that the relevant organisational security policies of the Platform-ST are not contradictory to those of the Composite-ST.

35 This work unit relates to *Step 3* of the *Application Notes* above. The evaluator compares the *relevant* organisational security policies (i.e. being mapped to the relevant TOE security objectives, cf. the work unit

ASE_COMP.1-5) of the Platform-ST with those of the Composite-ST and determines whether they are not contradictory.

36 Beyond it, a special organisational security policy OSP.Composite could be formulated within the Composite-ST, e.g. 'The application is running on a certified platform and compatible with it'. Then the developer can define the special security objectives for the TOE and its environment exactly reflecting the conditions and restrictions of the certification report of the underlying platform.

37 The result of this work unit shall be integrated to the result of ASE_SPD.1.3C/ ASE_SPD.1-3.

ASE_COMP.1-8 The evaluator ***shall examine*** the statement of compatibility to determine that the relevant organisational security policies of the Platform-ST are not contradictory to the threats of the Composite-ST and vice versa.

38 This work unit relates to *Step 3* of the *Application Notes* above. The evaluator compares the *relevant* organisational security policies (i.e. being mapped to the relevant TOE security objectives, cf. the work unit ASE_COMP.1-5) of the Platform-ST with the threats of the Composite-ST and determines whether they are not contradictory.

39 An example for contradictive items: The organisational security policy of the Platform-ST "Cryptographic algorithms used shall be in accordance with international standards" is contradictory to the threat of the Composite-ST "An attacker discloses the secrets being used by the TOE proprietary cryptographic algorithm".

40 The result of this work unit shall be integrated to the result of ASE_SPD.1.3C/ ASE_SPD.1.3C.

ASE_COMP.1-9 The evaluator ***shall examine*** the statement of compatibility to determine that the list of the assumptions of the Platform-ST being significant for the Composite-ST is complete and consistent for the current composite TOE.

41 This work unit relates to *Step 3* of the *Application Notes* above. In order to determine which assumptions of the Platform-ST are *significant* for the Composite-ST the evaluator analyses the assumptions of the Platform-ST and their separation in the following groups:

– **IrPA**: The assumptions being not relevant for the Composite-ST, e.g. the assumptions about the developing and manufacturing phases of the platform.
– **CfPA**: The assumptions being fulfilled by the Composite-ST *automatically*. Such assumptions of the Platform-ST can always be assigned to the TOE security objectives of the Composite-ST. Due to this fact they will be fulfilled either by the Composite-SFR or by the

Composite-SAR automatically. To give an example, let there be an assumption A.Resp-Appl of the Platform-ST: 'All User Data are owned by Smartcard Embedded Software. Therefore, it must be assumed that security relevant User Data (especially cryptographic keys) are treated by the Smartcard Embedded Software as defined for the specific application context' and a TOE security objective OT.Key_Secrecy of the Composite-ST: 'The secrecy of the signature private key used for signature generation is reasonably assured against attacks with a high attack potential.' If the private key is the only sensitive data element, then the assumption A.Resp-Appl is covered by the TOE security objective OT.Key_Secrecy automatically.
– **SgPA**: The remaining assumptions of the Platform-ST belonging neither to the group *IrPA* nor *CfPA*. **Exactly this group makes up the *significant* assumptions for the Composite-ST**, which shall be included into the Composite-ST.

42 The result of this work unit shall be integrated to the result of ASE_SPD.1.4C/ ASE_SPD.1-4.

ASE_COMP.1-10 The evaluator ***shall examine*** the statement of compatibility to determine that the significant security objectives for the operational environments of the Platform-ST are not contradictory to those of the Composite-ST.

43 This work unit relates to *Step 3* of the *Application Notes* above. The *significant* security objectives for the operational environment of the Platform-ST are the security objectives for the operational environment being assigned to the assumptions classified as the group *SgPA* of the Platform-ST (cf. the work unit ASE_COMP.1-9).

44 In order to accomplish this work unit the evaluator compares the *significant* security objectives for the operational environment of the Platform-ST with those of the Composite-ST and determines whether they are not contradictory. If necessary, the *significant* security objectives for the operational environment of the Platform-ST shall be included into the Composite-ST and assigned to the assumptions from the group *SgPA*, cf. the work unit ASE_COMP.1-8. The inclusion is not necessary, if the Composite-ST already contains equivalent (or similar) security objectives (covering all relevant aspects).

45 Since assurance of the development and manufacturing environment of the platform is confirmed by the platform certificate, the respective platform-objectives, if any, belong to the group *IrPA*.

46 Assurance of development and manufacturing environment is usually completely addressed by the assurance class ALC, and, hence, requires no explicit security objective.

47 The result of this work unit shall be integrated to the result of ASE_OBJ.2.1C/ ASE_OBJ.2-1 and ASE_OBJ.2.3C/ ASE_OBJ.2-3.

2. Integration of composition parts and Consistency of delivery procedures (ALC_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the ALC class listed in the following table. The other activities of ALC class do not require composite-specific work units.

CC assurance family	Evaluation activity	Evaluation work unit	Composite-specific work unit
ALC_CMS	ALC_CMS.12C	ALC_CMS.1-2	ALC_COMP.1-1
ALC_DEL	ALC_DEL.1.1C	ALC_DEL.1-1	ALC_COMP.1-3
AGD_PRE	AGD_PRE.1.2C	AGD_PRE.1-4	ALC_COMP.1-2

NB: If the level of the assurance requirement chosen is higher than those identified in this table, the composite-specific work unit is also applicable.

ALC_COMP.1 Integration of the application into the underlying platform and Consistency check for delivery and acceptance procedures

Objectives

48 The aims of this activity are to determine whether

– the correct version of the application is installed onto/into the correct version of the underlying platform, and

– the delivery procedures of **Platform** and **Application Developers** are compatible with the acceptance procedure of the **Composite Product Integrator**.

Dependencies:

49 No dependencies.

Developer action elements:

ALC_COMP.1.1D

50 The developer shall provide components configuration evidence; cf. item #7 in Table 1, section 4.7.

ALC_COMP.1.2D

51 The developer shall provide an evidence for delivery and acceptance compatibility; cf. item #8 in Table 1, section 4.7.

Content and presentation of evidence elements:

ALC_COMP.1.1C

52 The components configuration evidence shall show that

(i) the evaluated version of the application has been installed onto / embedded into the certified version of the underlying platform and

(ii) the configuration parameters evidence shall show that configuration parameters prescribed by the Platform and Application Developers are actually being used by the Composite Product Integrator.

ALC_COMP.1.2C

53 The evidence for delivery and acceptance compatibility shall show that the delivery procedures of the Platform and Application Developers are compatible with the acceptance procedure of the Composite Product Integrator.

Evaluator action elements:

ALC_COMP.1.1E

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

ALC_COMP.1.2E

55 The evaluator shall confirm that the evidence for delivery compatibility is complete, coherent, and internally consistent.

Evaluator actions:

Action ALC_COMP.1.1E

ALC_COMP.1-1 The evaluator ***shall check*** the evidence that the evaluated version of the application has been installed onto / embedded into the correct, certified version of the underlying platform.

56 The *general* information of the CM capabilities is represented and has to be examined in the context of the assurance family ALC_CMC. The *special* composite evaluator activity is to check the evidence of the version correctness for both parts of the composite product.

57 For the underlying platform, the evaluator shall determine that the actual identification of the platform is commensurate with the respective data in the platform certificate.

58 For the application, the relevant task is trivial due to the fact that the **Composite Product Evaluator** has to perform this task in the context of the assurance family ALC_CMC.

59 Components identification evidence can be supplied in two different ways: *technical* and *organisational*. A technical evidence of version correctness is being generated by the composite product itself: the platform and the application return – in each case – strings containing unambiguous version numbers as answers to the respective commands. E.g. it can be the return string of a command or the hard copy of the Windows-Information (like ‘About’); in case of smart cards it can be an appropriate ATR.

60 A technical evidence of version correctness for hardware can also be supplied, if applicable, by reading off the unambiguous inscription on its

surface. Note that there are no physical indication existing on most smart cards microcontrollers.

61 Technical evidence is recommended to be provided.

62 An organisational evidence of version correctness is being generated by the **Composite Product Integrator** on the basis of his configuration lists containing unambiguous version information of the platform and the application having been composed into the final composite product.

63 For example, in case of smart cards it can be an acknowledgement statement (e.g. configuration list) of the integrated circuit[34] manufacturer to the embedded software[35] manufacturer containing the evidence for the versions of the chip, the embedded software and its pre-personalisation parameters[36].

64 Organisational evidence is always possible and, hence, shall be provided.

65 The result of this work unit shall be integrated to the result of ALC_CMS.1.2C/ ALC_CMS.1-2 (or the equivalent higher components if a higher assurance level is selected).

ALC_COMP.1-2 The evaluator ***shall examine*** the evidence for using configuration parameters to determine that the Composite Product Integrator uses the configuration parameters prescribed by the Platform and Application Developers.

66 The *general* information of the configuration parameters required is represented and has to be examined in the context of the assurance family AGD_PRE [1.2C]. The *special* evaluator activity is to examine the developer's evidence and to decide whether the **Composite Product Integrator** appropriately treats this *special subset* of the configuration parameters.

67 For example, for a Java Card as composite TOE, the Card Issuer has to set all parameters as prescribed by the Java Card Platform and the Applet Developers while installing the applet onto the Java Card platform; cf. Table 3, section 4.7.

68 The result of this work unit shall be integrated to the result of AGD_PRE.1.2C/AGD_PRE.1-4.

---

[34] -> underlying platform

[35] -> application

[36] Any data supplied by the embedded software manufacturer that is injected into the non-volatile memory by the integrated circuits manufacturer. These data are for instance used for traceability and/or to secure shipment between phases (cf. [Smartcard IC Platform Protection Profile, Version 1.0, July 2001, registration number BSI-PP-0002], sec. 8.7).

Action ALC_COMP.1.2E

ALC_COMP.1-3 The evaluator ***shall examine*** the evidence for compatibility of delivery interfaces to determine that delivery procedures of the Platform and Application Developers are compatible with the acceptance procedure of the Composite Product Integrator.

69 The *general* information of the delivery procedures is represented and has to be examined in the context of the assurance families ALC_DEL and AGD_PRE [1.1C]. The *additional* composite activity of the evaluator is to examine each delivery interface between the **Platform Developer** and the **Composite Product Integrator** on the one side and between the **Application Developer** and the **Composite Product Integrator** on the other side. As a result, the evaluator confirms or disproves the justification for delivery compatibility.

70 If there are no delivery interfaces between the **Platform** and **Application Developers** and the **Composite Product Integrator** or the assurance package chosen does not contain the family ALC_DEL (e.g. EAL1), this work unit is not applicable.

71 The result of this work unit shall be integrated to the result of ALC_DEL.1.1C/ ALC_DEL.1-1.

3. Composite design compliance (ADV_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the ADV class listed in the following table. The other activities of ADV class do not require composite-specific work units.

CC assurance family	Evaluation activity	Evaluation work unit	Composite-specific work unit
ADV_ARC	ADV_ARC.1.1E	ADV_ARC.1.1C/ ADV_ARC.1-1	ADV_COMP.1-1
ADV_IMP	ADV_IMP.1.1E	ADV_IMP.1.1C/ ADV_IMP.1-1	ADV_COMP.1-1
ADV_INT	ADV_INT.2.1E	ADV_INT.2.1C/ ADV_INT.2-1	ADV_COMP.1-2
ADV_TDS	ADV_TDS.1.2E	ADV_TDS.1-7	ADV_COMP.1-1

NB: If the level of the assurance requirement chosen is higher than those identified in this table, the composite-specific work unit is also applicable.

ADV_COMP.1 Design compliance with the platform certification report, guidance and ETR_COMP

Objectives

72 The aim of this activity is to determine whether the requirements on the application, imposed by the underlying platform, are fulfilled in the composite product.

Application notes

73 The requirements on the application, imposed by the underlying platform, can be formulated in the relevant certification report (e.g. in form of constraints and recommendations), user guidance and ETR_COMP (in form of observations and recommendations) for the platform. The developer of the composite product shall regard each of these sources, if available (cf. Table 2, section 4.7), and implement the composite product in such a way that the applicable requirements are fulfilled.

74 The TSF of the composite product is represented at various levels of abstraction in the families of the development class ADV. Experiential, the appropriate levels of design representation for examining, whether the requirements of the platform are fulfilled by the composite product, are the TOE design (ADV_TDS), security architecture (ADV_ARC) and the implementation (ADV_IMP). In case, these design representation levels are not available (e.g. due to the assurance package chosen is EAL1), the current activity is not applicable (see the next paragraph for the reason).

75 Due to the definition of the composite TOE (cf. section 2.1 ‘Definitions’) the interface between the underlying platform and the application is the *internal* one, hence, a functional specification (ADV_FSP) as representation level is not appropriate for analysing the design compliance.

76 Security architecture ADV_ARC as assurance family is dedicated to ensure that integrative security services like domain separation, self-protection and non-bypassability properly work. It is impossible and not the sense of the composite evaluation to have an insight into the architectural internals of the underlying platform (it is a matter of the platform evaluation). What the **Composite Evaluator** has to do in the context of ADV_ARC is
(i) to determine whether the application uses services of the underlying platform **within its own Composite-ST** to provide domain separation, self-protection, non-bypassability and protected start-up; if no, there is no further composite activities for ADV_ARC; if yes, then
(ii) the evaluator has to determine, whether the application uses these platform-services in an appropriate/secure way (please refer to the platform user guidance, cf. item #3 in Table 1, section 4.7.).

77 Since consistency of the composite product security policy has already been considered in the context of the Security Target in the assurance family ASE_COMP (see page 51 above), there is no necessity to consider non-contradictoriness of the security policy model (ADV_SPM) of the composite TOE and the security policy model of the underlying platform.

Dependencies:

78 No dependencies.

Developer action elements:

ADV_COMP.1.1D

79 The developer shall provide a design compliance justification; cf. item #6 as well as items #3, #4, #5 in Table 1, section 4.7..

Content and presentation of evidence elements:

ADV_COMP.1.1C

80 The design compliance justification shall provide a rationale for design compliance – on an appropriate representation level – of how the requirements on the application, imposed by the underlying platform, are fulfilled in the composite product.

Evaluator action elements:

ADV_COMP.1.1E

81 The evaluator shall confirm that the rationale for design compliance is complete, coherent, and internally consistent.

Evaluator actions:

Action ADV_COMP.1.1E

ADV_COMP.1-1 The evaluator ***shall examine*** the rationale for design compliance to determine that all applicable requirements on the application, imposed by the underlying platform, are fulfilled by the composite product.

82 In order to perform this work unit the evaluator shall use the rationale for design compliance as well as the TSF representation on the ADV_TDS, ADV_ARC and ADV_IMP levels on the one side and the input of the platform developer in form of the certification report, guidance and ETR_COMP on the other side. The evaluator shall analyse which platform requirements are applicable for the current composite product. The evaluator shall compare each of the applicable requirements with the actual specification and/or implementation of the composite product and determine, for each requirement, whether it is fulfilled. As result, the evaluator confirms or disproves the rationale for design compliance.

83 For example, platform guidance may require the application to perform a special start-up sequence testing the current state of the platform and initialising its self-protection mechanisms. Such information might be found in the description of secure architecture ADV_ARC of the composite TOE; see also the *Application Note* above.

84 The appropriate representation level (ADV_TDS, ADV_ARC and/or ADV_IMP), what the analysis is being performed on, can be chosen and mixed flexibly depending on the concrete composite TOE and the requirement in question. Where it is not self-explaining, the evaluator shall justify why the representation level chosen is appropriate.

85 The evaluator activities in the context of this work unit can be spread over different single evaluation aspects (e.g. over ADV_TDS and ADV_IMP). In this case the evaluator performs the partial activity in the context of the corresponding single evaluation aspect. Then the notation for this work unit shall be ADV_COMP.1-1-TDS, ADV_COMP.1-1-ARC and ADV_COMP.1-1-IMP, respectively.

86 If the assurance package chosen does not contain the families ADV_TDS, ADV_ARC or ADV_IMP (e.g. EAL1), this work unit is not applicable (cf. *Application Note* above).

87 The result of this work unit shall be integrated to the result of ADV_TDS.1−2E/ ADV_TDS.1-7, ADV_ARC.1.1E/ ADV_ARC.1.1C/ ADV_ARC.1-1, ADV_IMP.1.1E/ ADV_IMP.1.1C/ ADV_IMP.1-1 (or the equivalent higher components if a higher assurance level is selected).

ADV_COMP.1-2 The evaluator ***shall check*** the TSF internals of the composite TOE to determine that they do not contradict any design requirement imposed by the underlying platform.

88 The TSF internals are represented and evaluated in the context of the assurance family ADV_INT. The evaluator shall compare the internal structure of the TSF with the design requirements of the platform and search for obvious contradictions.

89 If there are no requirements of the platform concerning the TSF internal structure or the assurance package chosen does not contain the family ADV_INT, this work unit is not applicable.

90 The result of this work unit shall be integrated to the result of ADV_INT.2.1E / ADV_INT.2.1C/ ADV_INT.2-1 (or the equivalent higher components if a higher assurance level is selected).

4. Composite functional testing (ATE_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the ATE class listed in the following table. The other activities of ATE class do not require composite-specific work units.

CC assurance family	Evaluation activity	Evaluation work unit	Composite-specific work unit
ATE_COV	ATE_COV.1.1C	ATE_COV.1-1	ATE_COMP.1-1
	ATE_COV.1.1C	ATE_COV.1-1	ATE_COMP.1-2
ATE_FUN	ATE_FUN.1.2C	ATE_FUN.1-3	ATE_COMP.1-1
ATE_IND	ATE_IND.1.2E	ATE_IND.1-5	ATE_COMP.1-3

NB: If the level of the assurance requirement chosen is higher than those identified in this table, the composite-specific work unit is also applicable.

ATE_COMP.1 Composite product functional testing

Objectives

91 The aim of this activity is to determine whether composite product *as a whole* exhibits the properties necessary to satisfy the functional requirements of its Security Target.

Application notes

92 A composite product can be tested *separately* and *integrative*. Separate testing means that the platform and the application are being tested independent of each other. A lot of tests of the platform may have been performed within the scope of its accomplished evaluation. The application may be tested on a simulator or an emulator, which represent a virtual machine.
Integration testing means that the composite product is being tested as it is: the application is running on the platform.

93 Behaviour of implementation of some SFRs can depend on properties of the underlying platform as well as of the application (e.g. correctness of the measures of the composite product to withstand a side channel attack or correctness of the implementation of tamper resistance against physical attacks). In such a case the SFR implementation shall be tested on the final composite product, but not on a simulator or an emulator.

94 This activity focuses exclusively on testing of the composite product *as a whole* and represents merely *partial efforts* within the general test approach being covered by the assurance ATE. These integration tests shall be specified and performed, whereby the approach of the standard[37] assurance families of the class ATE shall be applied.

[37] i.e. as defined by CEM

95 - A correct behaviour of the Platform-TSF being relevant for the Composite-ST (corresponding to the group *RP_SFR* in the work unit 0 above), and
- absence of exploitable vulnerabilities (sufficient effectiveness) in the context of the Platform-ST
are confirmed by the valid Platform Certificate, cf. chapter 6 above.

Dependencies:

96 No dependencies.

Developer action elements:

ATE_COMP.1.1D

97 The developer shall provide a set of tests as required by the assurance package chosen.

ATE_COMP.1.2D

98 The developer shall provide the composite TOE for testing.

Content and presentation of evidence elements:

ATE_COMP.1.1C

99 Content and presentation of the specification and documentation of the *integration* tests shall correspond to the standard[38] requirements of the assurance families ATE_FUN and ATE_COV.

ATE_COMP.1.2C

100 The composite TOE provided shall be suitable for testing.

Evaluator action elements:

ATE_COMP.1.1E

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

ATE_COMP.1.2E

102 The evaluator shall specify, perform and document a set of own *integration* tests to confirm that the composite TOE operates as specified.

Evaluator actions:

Action ATE_COMP.1.1E

ATE_COMP.1-1 The evaluator ***shall examine*** that the developer performed the *integration* tests for all SFRs having to be tested on the composite product as a whole.

103 In order to perform this work unit the evaluator shall analyse, for each SFR, whether it directly depends on security properties of the platform and of the application. Then the evaluator shall verify that the *integration* tests performed by the developer cover at least all such SFRs.

104 If the assurance package chosen does not contain the families ATE_FUN and ATE_COV (e.g. EAL1), this work unit is not applicable.

[38] i.e. as defined by CEM

105 The result of this work unit shall be integrated to the result of ATE_COV.1−1C/ ATE_COV.1-1 and ATE_FUN.1.2C/ ATE_FUN.1-3 (or the equivalent higher components if a higher assurance level is selected).

106

Action ATE_COMP.1.2E

ATE_COMP.1-2 The evaluator ***shall determine*** the minimal amount of the *integration* tests being necessary for the current composite evaluation.

107 The evaluator shall perform the following steps:

1) The evaluator determines the share of the platform part of the TOE in enforcing of the Composite-ST. In order to do this, the evaluator refers to the TOE design (ADV_TDS and ADV_ARC might possess an appropriate representation level for this purpose) as well as to the Composite-ST and lists all Composite-SFRs using the security services of the platform[39]. Hereby the evaluator shall understand/document, for each such Composite-SFR, what concretely the platform does (so called *platform's share*).

For example, there is a Composite-SFR FCS_CKM.1 fulfilled by the TSF-portion 'Key Generation'. The share of the platform in implementing this SFR is generating random numbers for key generation.

2) The evaluator checks, for each such Composite-SFR, whether the *platform's share* has been covered by the Platform Certificate; in order to do this the evaluator refers to the platform user guidance, ETR_COMP and the platform certification report.

For our example with the random number generator (RNG), the evaluator might find an advice in ETR_COMP that (i) the RNG is regularly being tested (online tests), (ii) its behaviour concerning power consumption analysis was evaluated and (iii) a sufficient (for generation of static keys) random numbers quality was confirmed by the Platform Certificate. In such a case the evaluator can decide that no *integration* tests are necessary for FCS_CKM.1/Key Generation[40].

The next example represents a situation, where additional *integration* tests are necessary. There let be a Composite-SFR FPT_EMSEC.1 (electromagnetic emanation) fulfilled by the following TSF-portions:

- 'User Authentication'; the platform's share in implementing this SFR is

[39] Such security services of the platform are represented by the group *RP_SFR, cf.* the work unit 0 above

[40] Of course, the evaluator might (and, perhaps, would) decide to test 'Key Generation'-TSF as an integration test on the final TOE. A methodological reason for this test would then be rather a general check of the respective functionality, but not the confirmation of security behaviour of the platform-RNG.

scrambling the reference authentication data stored in the TOE;
- 'Key Generation'; the platform's share in implementing this SFR is disguising information about the value of the key being generated while observing power consumption;
- 'Signature Creation'; the platform's share in implementing this SFR is disguising information about the value of the signature key being used while observing power consumption;

For scrambling, the evaluator might find the advice in ETR_COMP that the platform scrambling engine has been evaluated, but it must be correctly initialised by the application. Hence, the evaluator has to specify an integration test for initialising the scrambling engine.
For disguising during key generation, the evaluator might find no advices in ETR_COMP. Hence, the evaluator has to assume that this behaviour of the platform was not included in consideration. Therefore, the evaluator has to specify an integration test for power consumption while generating the signature keys.
For disguising during signature creation, the evaluator might find an advice in ETR_COMP that DPA and SPA on RSA have been evaluated, but no advice of Timing on RSA. Hence, the evaluator has to assume that the Timing-on-RSA behaviour of the platform was not included in consideration. Therefore, the evaluator has to specify an integration Timing-on-RSA test for power consumption while using the signature key.

3) The evaluator refers to ETR_COMP and checks for any explicit requirements for performing tests in the context of the composite evaluation. Such explicit requirements might sound like 'Such test has to be performed during the SW evaluation'.

All platform tests being necessary for the current composite evaluation, but not covered by the Platform Certificate, and all tests explicitly required by ETR_COMP, encompass the minimal amount of the integration tests being necessary for the current composite evaluation.

This work unit is the complementary part to the work unit ASE_COMP.1-1 In ASE_COMP.1-1 the evaluator determines, on which part of the Platform-ST the Composite-ST can rely (the group *RP_SFR*); in the current work unit the evaluator determines, whether the Composite-ST can also rely on the platform's *functional* behaviour being not covered by the Platform-ST.

108 The result of this work unit shall be integrated to the result of ATE_COV.1.1C/ ATE_COV.1-1 (or the equivalent higher components if a higher assurance level is selected).

ATE_COMP.1-3 The evaluator ***shall perform*** the standard[41] evaluator actions in the context of the assurance family ATE_IND on the set of the *integration* tests using the composite product as a whole.

109 The set of the *integration* tests for this activity shall embrace at least the minimal amount of the integration tests as determined in the previous work unit.

110 The result of this work unit shall be integrated to the result of ATE_IND.1.2E/ ATE_IND.1-5) (or the equivalent higher components if a higher assurance level is selected).

[41] i.e. as defined by CEM

5. Composite vulnerability assessment (AVA_COMP)

The composite-specific work units defined in this chapter are intended to be integrated as refinements to the evaluation activities of the AVA class listed in the following table. The other activities of AVA class do not require composite-specific work units.

CC assurance family	Evaluation activity	Evaluation work unit	Composite-specific work unit
AVA_VAN	AVA_VAN.1.3E	AVA_VAN.1-5	AVA_COMP.1-1
	AVA_VAN.1.3E.	AVA_VAN.1-6	AVA_COMP.1-2
	AVA_VAN.1.3E	AVA_VAN.1-7	AVA_COMP.1-2
	AVA_VAN.1.3E	AVA_VAN.1-8	AVA_COMP.1-2

NB: If the level of the assurance requirement chosen is higher than those identified in this table, the composite-specific work unit is also applicable.

AVA_COMP.1 Composite product vulnerability assessment

Objectives

111 The aim of this activity is to determine the exploitability of flaws or weaknesses in the composite TOE *as a whole* in the intended environment.

Application notes

112 This activity focuses exclusively on vulnerability assessment of the composite product *as a whole* and represents merely *partial efforts* within the general approach being covered by the standard[42] assurance family of the class AVA: AVA_VAN.

113 The results of the vulnerability assessment for the underlying platform represented in the ETR_COMP can be reused, if they are up to date and all composite activities for correctness – ASE_COMP.1, ALC_COMP.1, ADV_COMP.1 and ATE_COMP.1 – are finalised with the verdict PASS.

114 Due to composing of the platform and the application a new quality arises, which can cause additional vulnerabilities of the platform which might be not mentioned in the ETR_COMP.

Dependencies:

115 No dependencies.

Developer action elements:

AVA_COMP.1.1D

116 The developer shall provide the composite TOE for penetrating testing.

Content and presentation of evidence elements:

AVA_COMP.1.1C

117 The composite TOE provided shall be suitable for testing *as a whole*.

[42] i.e. as defined by CEM

Evaluator action elements:

AVA_COMP.1.1E

118 The evaluator shall conduct penetration testing of the composite product *as a whole* building on evaluator's own vulnerability analysis, to ensure that the vulnerabilities being relevant for the Composite-ST are not exploitable.

Evaluator actions:

Action AVA_COMP.1.1E

AVA_COMP.1-1 The evaluator ***shall examine*** the results of the vulnerability assessment for the underlying platform to determine that they can be reused for the composite evaluation.

119 The results of the vulnerability assessment for the underlying platform are usually represented in the ETR_COMP. They can be reused, if they are up to date and all composite activities for correctness – ASE_COMP.1, ALC_COMP.1, ADV_COMP.1 and ATE_COMP.1 – are finalised with the verdict PASS. The evaluator shall also consider the relevant determinations in the Platform Certification Report. For validity of the platform security certificate please refer to chapter 6 above.

120 The result of this work unit shall be integrated to the result of AVA_VAN.1.3E/ AVA_VAN.1-5 (or the equivalent higher components if a higher assurance level is selected).

AVA_COMP.1-2 The evaluator ***shall specify, conduct and document*** penetration testing of the composite product *as a whole*, using the standard approach of the assurance family AVA_VAN.

121 If the correctness-related activities – ASE_COMP.1, ALC_COMP.1, ADV_COMP.1 and ATE_COMP.1 – are finalised with the verdict PASS and the certificate for the platform covers all security properties needed for the composite product, composing of the platform and the application must not create additional vulnerabilities of the platform.

122 If the evaluator determined that composing of the platform and the application creates additional vulnerabilities of the platform[43], a contradiction to the verdict PASS for the correctness activities (see paragraph 113 above) has to be supposed or the certificate for the platform does not cover all security properties needed for the current composite product.

123 The result of this work unit shall be integrated to the result of AVA_VAN.1.3E/ AVA_VAN.1-6, AVA_VAN.1-7, AVA_VAN.1-8 (or the equivalent higher components if a higher assurance level is selected).

[43] i.e. not mentioned in the ETR_COMP

Appendix 2: ETR for composite evaluation template

The related document "JIL-ETR-template-for-composition" shall be used as a template by the **Platform Developer** to issue the ETR_COMP. Please note that the layout can be customized according to the evaluation facilities company standard, but the contents and structure are mandatory.