LOGO

<NAME OF ITSEF COMPANY>

ETR for composite evaluation
TD SC & SD

<Name product>

TEMPLATE

COMMERCIAL IN CONFIDENCE

Product	<Name product>
Developer	
Sponsor	
Certification Body	
Certificate reference	
Evaluation facility	
Evaluator(s)	

Note from SOGIS-MRA: This document is an output when applying the composite model and its concept of ETR for composite evaluation as outlined in [COMP] specifically for Smartcard and similar devices type of products. In the technical domain SC & SD, the 'base component' equals the platform and the 'dependent component' is the application.

In order to support the efficiency of this concept and to be able to maintain the platform product assurance, the ITSEF using this document shall inform the platform certification body (possibly via their oversight body) if failures or vulnerabilities of the platform (e.g. vulnerabilities due to improved attack methods or techniques), have been detected within the course of the composite product evaluation.

Reference/version	*<Reference and version of the document>*
Date	*<Date of the document>*

Document information

Document and project identification

Name	Value
Document title	
Reference/version	
CC version	
Evaluation level	
Developer	
Sponsor	
Certification Body	
Certificate reference	
Evaluation facility	

Version history

Version	Date	Nature of modification	Page

Edition

<Name of the document writers>

Approval

<Name and title of the approver, date of approval, visa>

Distribution

<Distribution list>

- Certification body
- Sponsor
- Developer
- Observer

Confidentiality and copyright notice

<Any specific mention to confidentiality rules and copyright notice>

Table of contents

1. Introduction

1.1. Objective of the document

The standard Evaluation Technical Report [ETR] contains proprietary information that cannot be made public. This document is compiled from the [ETR] in order to provide sufficient information for composite evaluation with the certified TOE <Name product>. It contains information from the TOE evaluation needed for composite evaluation and should enable the reader to understand the threats and the effectiveness of countermeasures. This document was written according to the referenced document [COMP].

The targeted audience are ITSEF that conduct composite evaluation based on <Name product>.

1.2. Product identification

The evaluated revision of the product is: <Name product>.
<Add any useful detail to the product main reference, in order to provide all necessary information to identify clearly the product during the composite evaluation:

- *Identification of the hardware part,*
- *Identification of all software libraries included,*
- *Identification of possible software platform included>*

These references are provided with the following rules:
<Describe the manufacturer rules to understand the references given: commercial reference, technical reference: possible firmware reference, software platform reference, software libraries references, hardware part references (identification of the production site if more than one is used, etc…), identification of the complete configuration list [CONF] etc…>

The way to check the revision of the product is described in chapter 3.2.

The list of guidance to use with the product in its certified configuration is given in Annex 1 ([AGD-X]).

<<it is noted that after a certification maintenance is performed, resulting in an update of the guidance, an evaluator is made aware of changes by listing versions here in the ETR_COMP, as well in a possibly change of the ST and/or Certification report>>.

1.3. Evaluation results and certification summary

The content given in this report is a result of the product <Name product> evaluation as specified in the <Name product> security target [ST].
<Add possible comments and history about re-evaluation and references of the previous certified product, previous ETR and task re-use.>

The evaluation tasks have been performed in compliance to Common Criteria [CC] and its methodology [CEM] at level <*EAL4/5/6 augmented*>. The following table details the selected <*EAL4/5/6 augmentations*>:
<*Add the list of assurance components that are augmented compared to the assurance level defined in Common Criteria like the following table.*>

Assurance component	
EAL4	Methodically designed, tested, and reviewed
+ ALC_DVS.2	Sufficiency of security measures
+ AVA_VAN.5	Advanced methodical vulnerability Analysis

Table 1 - Assurance component for CC:2022 evaluation

<*For the assurance components higher than EAL4 level, the evaluators have used <proprietary methods validated by the evaluation authorities>.*>

The evaluation has been performed also with the help of the following Common Criteria supporting documents:

- "The Application of CC to Integrated Circuits" (cf. [JIL IC]),
- "Application of Attack Potential to Smartcards and Similar Devices" (cf. [JIL AP]),
- <*other certification body specific document*>

The evaluation has been performed also with the help of the following JIL supporting document(s):

- "Attack Methods for Smartcards and Similar Devices" (cf. [JIL AM])
- <*other certification body specific documents*>

The product was certified by the <*identification of the certification body*> under the reference <*reference of the certificate/certification report*> (cf. [CERTIF]) on the <*date of certification*>.

The product shall be used with its guidance identified in Annex 1 under the reference [AGD-X].

The delivery procedures of the Platform Developer identified under the reference [DEL] and detailed in chapter 4 shall be followed by the Application Developer.

1.4. Contact

1.4.1. Evaluator

<*Possible introduction to the evaluator, with reference to the accreditation and/or licensing number from the scheme.*>

1.4.2. Sponsor and developer

<*Possible introduction to the sponsor and developer, with address and contact for product and certification information.*>

1.4.3. Certification Body

<Possible introduction to the Certification Body, with address and contact for certification information.>

2. Platform Design

This section of the ETR_COMP shall provide a high-level description of the platform 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 the parts developed and added by the Composite Product Developer. This shall include a list of security mechanisms of the platform covered by the platform evaluation.

2.1. General conception

The product is a <*single chip micro-controller unit / microcontroller with software platform*> designed by <*developer*> and built in 0.XXµm <*details on the type of technology shall be included*>.

<*EXAMPLES*>

2.2. Example of an IC architecture description

The guidance documentation (Data Sheet, User Guidance Manual, etc.) provided as part of the platform was subject of the platform evaluation. Therefore, the content is considered to provide sufficient information for a developer using this platform and the composite evaluator. Since the ETR_COMP shall not include information already provided within other documents this information is not reproduced here. The following aspects may not be described in sufficient detail or may not be obvious from the documentation and shall be therefore included here:

- Security mechanisms not described in the guidance documentation
- Configurable security mechanisms
- Separation of SFR non-interfering parts

These aspects are detailed in the following:

1. The platform may implement security mechanisms not described in the guidance documentation since they can neither be configured nor enabled/disabled by the user of the platform. This can comprise active shielding techniques, sensors, masking, etc. disabling the device or forcing specific actions. These security mechanisms support the resistance against attacks and need to be known by the composite evaluator to interpret and assess the results of the penetration testing of the composite product.
2. Many platforms can be customised to some extent during the wafer testing. This configuration can include the enabling or disabling of: External interfaces, dedicated security mechanisms supporting the resistance of specific components or preventing specific attacks as well as security mechanisms like coprocessors implementing Security Functional Requirements claimed in the Security Target. This section of the ETR_COMP shall include an overview of the configuration options that may have an impact for the platform evaluation. In combination with the description provided in sections 3.1 and 3.2 the composite evaluator shall be

able to determine the configuration of the specific platform used to build the composite TOE.

3. The platform may comprise components that are categorized as SFR non-interfering because they do not provide or support functionality specified by the Security Functional Requirements defined in the Security Target. This can comprise hardware components as well as parts of the IC Dedicated Software (firmware). This section of the ETR_COMP shall summarize the separation of the SFR non-interfering components to support the assessment for the composite product. As well as identification of modules that were considered non-TSF.

4. Limitations to the use of external interfaces/TSFI that are provided to the developer but upon usage do not lead to a certified composite TOE shall also be explained in this section.

2.2.1. Borders of the evaluation with regard to the module architecture and interfaces

Usually, every module of a hardware platform is rated security enforcing as the design is implemented by a synthesis process with related tooling following defined security and layout constraints. Parts of the synthesised hardware modules implement and constitute dedicated security mechanisms while others just contribute or support to a security mechanism. However, the hardware platform may also include SFR non-interfering components that are less protected or not protected by the supporting security mechanisms of the hardware.

Firmware delivered as part of the hardware platform may implement or support security mechanisms or may be SFR non-interfering.

The SFR non-interfering modules have been evaluated only with regard to functional verification and not in the context of resistance to attackers, as these modules neither represent a worthwhile target nor potential attacks lead to exploitable scenarios.

The following table lists the assignment of the modules whether they were considered in the resistance rating and the interfaces of each module.

The interface definition is as follows:

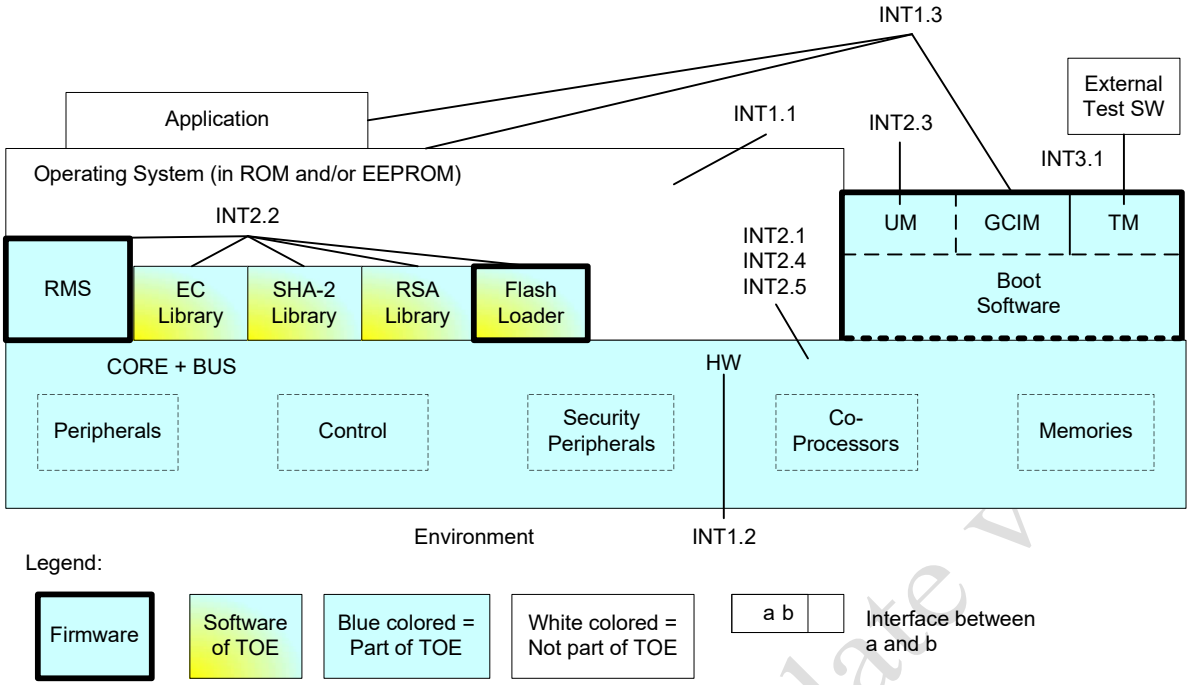
INT1.1 Physical Interface of the TOE
INT1.2 Electrical Interface of the TOE
INT1.3 Data Interface of the TOE
INT2.1 Instruction Set of the CPU
INT2.2 API Instructions
INT2.3 Interface to the Boot Software
INT2.4 Special Function Registers
INT2.5 Crypto Instruction Set
INT3.1 Interface of the Test Mode to the environment

Subsystem/Module	Assignment	Resistance tested	Interfaces
Core			
Dual CPU	Security enforcing	Yes	INT1.1, INT1.2, INT1.3, INT2.1, INT2.2, INT2.4
CACHE	Security enforcing	Yes	INT1.1, INT1.2, INT1.3, INT2.1, INT2.2

Subsystem/Module	Assignment	Resistance tested	Interfaces
MED with EDU	Security enforcing	Yes	INT1.1, INT1.2, INT1.3, INT2.1, INT2.2, INT2.4
MMU	Security enforcing	Yes	INT1.1, INT1.2, INT1.3, INT2.1, INT2.2, INT2.3, INT2.4
Memories			
ROM	Security enforcing	Yes	INT1.1, INT1.2, INT1.3, INT2.1, INT2.2, INT2.4
RAM	Security enforcing	Yes	INT1.1, INT1.2, INT1.3, INT2.1, INT2.4
Flash EEPROM	Security enforcing	Yes	INT1.1, INT1.2, INT1.3, INT2.1, INT2.4
Memory Bus	Security enforcing	Yes	INT1.1, INT1.2, INT1.3
Peripheral Bus	Security enforcing	Yes	INT1.1, INT1.2, INT1.3
Computing Peripherals			
Asymmetric Crypto Co-Processor	Security enforcing	Yes, with optional cryptographic libraries. The module implements no countermeasures against FI and SCA.	INT1.1, INT1.2, INT1.3, INT2.1, INT2.2
Symmetric Crypto Co-Processor	Security enforcing	Yes	INT1.1, INT1.2, INT1.3, INT2.1, INT2.2, INT2.4
CRC	Security enforcing	Yes, user obligations	INT1.1, INT1.2, INT1.3
Hash-Module	Security enforcing	Yes, user obligations	INT1.1, INT1.2, INT2.2, INT2.4
PRTNG	Security enforcing	Yes, with <*national regulation reference*> compliance test	INT1.1, INT1.2, INT1.3, INT2.1, INT2.2, INT2.4
DRNG	Security enforcing	Yes, no quality metric claimed	INT1.1, INT1.2, INT1.3, INT2.2, INT2.4
System Peripherals			
Chip Reset	Security enforcing	Yes	INT1.1, INT1.2, INT2.1, INT2.2, INT2.4
IMM	Security enforcing	Yes	INT1.1, INT1.2, INT2.1, INT2.2, INT2.4

Subsystem/Module	Assignment	Resistance tested	Interfaces
UMSLC	Security enforcing	Yes	INT1.1, INT1.2, INT2.1, INT2.2, INT2.4
Test Controller	Non-interfering	No, disabled in User Mode	INT1.1, INT1.2
CLKU	Security enforcing	Yes	INT1.1, INT1.2, INT2.4
Standard Peripherals			
Timers and Watchdogs	Security enforcing	Yes	INT1.1, INT1.2, INT1.3, INT2.1, INT2.2, INT2.4
ITP (Interrupt/Peripheral Controller)	Security enforcing	Functionally verified, no scenario	INT1.1, INT1.2, INT2.4
ISO interface	Security enforcing	Functionally verified, no scenario	INT1.1, INT1.2, INT1.3, INT2.4
UART	Security enforcing	Functionally verified, no scenario	INT1.1, INT1.2, INT2.4
USB	Security enforcing	Functionally verified, no scenario	INT1.1, INT1.2, INT2.4
IIC (Inter IC Interface)	Security enforcing	Functionally verified, no scenario	INT1.1, INT1.2, INT2.4
SSC (Synchronous Serial Controller)	Security enforcing	Functionally verified, no scenario	INT1.1, INT1.2, INT2.4
Radio Frequency Interface	Security enforcing	Functionally verified, no scenario	INT1.1, INT1.2, INT2.4
GPIO	Security enforcing	Functionally verified, no scenario	INT1.1, INT1.2, INT2.4
Boot Software	Security enforcing	Yes, attacks plus source code review	INT1.3, INT2.1, INT2.3
Resource Management System (Routines)	Security enforcing	Yes, attacks plus source code review	INT2.1, INT2.2
Flash Loader	Security enforcing	Yes, attacks plus source code review	INT2.2
Cryptographic Library RSA	Security enforcing	Yes, attacks plus source code review	INT2.1, INT2.2
Cryptographic Library EC	Security enforcing	Yes, attacks plus source code review	INT2.1, INT2.2
Cryptographic Library SHA-2	Security enforcing	Yes, attacks plus source code review	INT2.1, INT2.2

The following graph visualizes the coverage and interfaces of the TOE and clearly marks what belongs to the TOE:

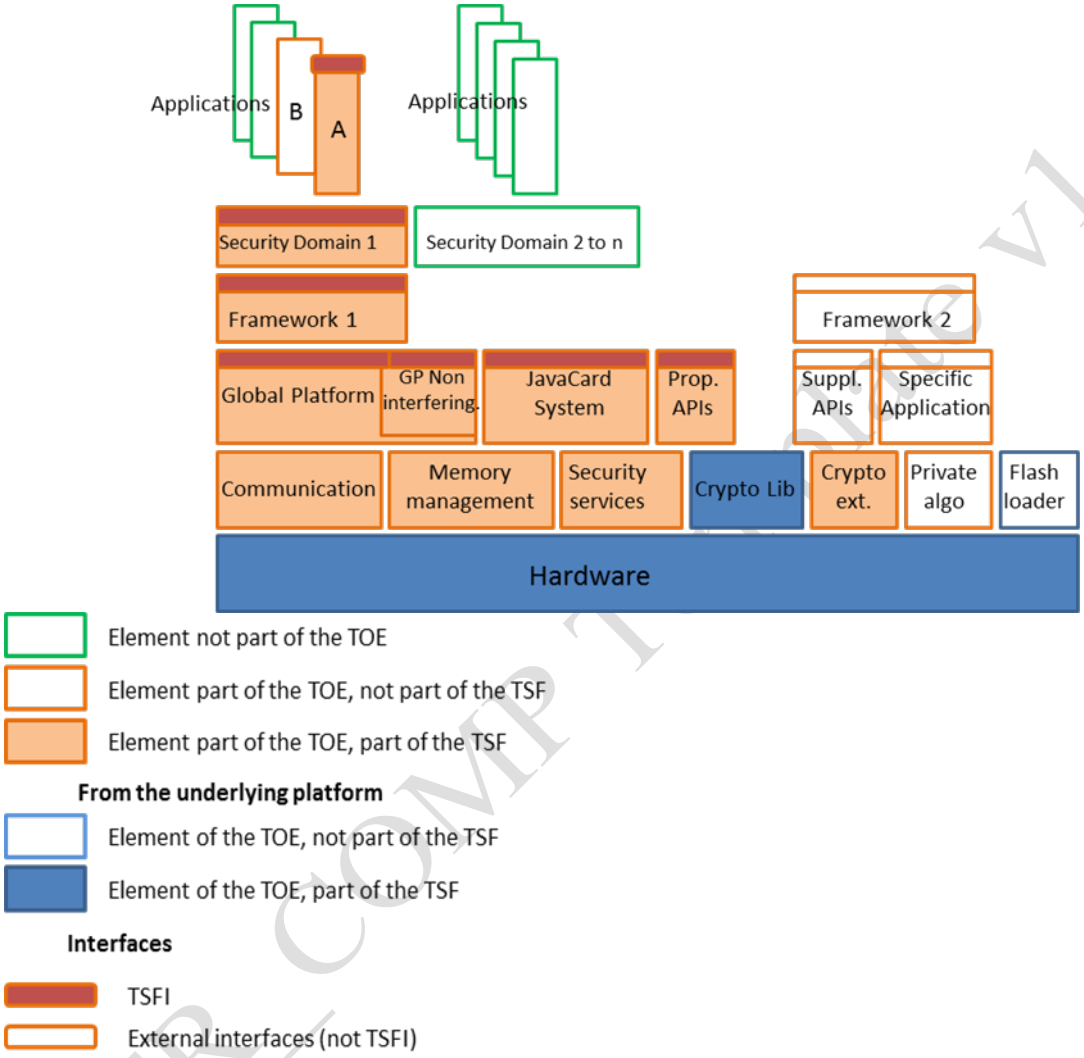


The symmetric crypto co-processor module implementing TDES and AES is included in the TOE. Only AES related TSFI are in the certification scope.

2.3. Example of an embedded software platform

2.3.1. Description

The following figure is a scheme showing the platform TOE and its internal decomposition in elements that are part of the TSF and elements that are not. The TOE interfaces are also represented.



The IC included in the TOE is compliant with the [PP-BSI-0084]. It includes a crypto library and a Flash loader used for software loading but inactivated for end user usage.

The TOE includes an OS layer with the following functionalities:

- Communication management
- Memory management
- Security services

An extension of the crypto library that is part of the TSF:

- A private crypto algorithm that doesn't implement security functions corresponding to any SFR of the platform Security Target and running in a specific mode (in a separated domain)

The TOE includes the Java Card Virtual Machine (JCVM), the Java Card Runtime Environment (JCRE) and the Java Card Application Programming Interface (JCAPI). This set is also named Java Card System. It is compliant with the standard xxx.

The TOE also includes other APIs that are not defined in the Java Card standard:

- Proprietary APIs: to provide enhanced security to the applications
- Supplemental APIs that do not implement security functions corresponding to any SFR of the platform Security Target and are running in a separate domain from the rest of the TSF

The TOE is compliant with the Global Platform standard XXX which provides a set of APIs and technologies to perform in secure way the operations involved in the management of the security domains and applications hosted by the card. This element manages the downloading and installing of applications on the platform. This element also provides the means for the applications to communicate with the external world on a standard basis.

The following GP functionalities are present within the TOE and implement security functions that have been evaluated:

- Card content loading
- Installation of Security Domains and Extradition
- DAP support and Mandated DAP support
- DAP calculation with asymmetric cryptography
- SCP02, SCP03 and SCP80 support
- Trusted Path privilege
- Delegated Management privilege
- Attribution of Authorized Management privilege only to Secure Domain 1

The following functionalities are present within the TOE but they do not implement security functions corresponding to any SFR. They are grouped in the element named GP non-interfering.

- Logical channels
- Support of contactless services
- Global PIN management
- Post-issuance personalization of Security Domains
- Application personalization

The TOE does not implement security functions corresponding to any SFRs defined in the platform Security Target and are running in a separate domain from the rest of the TSF.

The Security Domain 1 contributes to the administration of the card and, as such, it manages some aspect of the overall security.

Application A, which is a Java Card applet linked to Security Domain 1, offers specific security services to the external user that are described by SFRs of the platform Security Target.

Application B offers services during the construction of the TOE but is deactivated in the operational phase (after delivery).

2.3.2. Borders of the evaluation with regard to the architecture and interfaces

The platform TOE contains all the elements that must be evaluated with regard to the vulnerability analysis. Certain elements (sub-systems / modules) implement functions that are part of the realization of SFRs (SFR-enforcing, SFR-supporting) and as such are evaluated in terms of correctness. Other elements contain security non-interfering functionalities that have no role in the realization of the SFRs but are likely part of the TSF because if compromised it could compromise the correct operation of an SFR by virtue of its privileged running mode. They have to be taken into account for the vulnerability analysis.

It is possible that some elements are part of the TOE but are not security relevant, meaning they do not contribute to preserve security of the TOE as expressed by the SFRs and requirements for domain separation and non-bypassability. They are out of the scope of the TSF as far as the isolation property is resistant to attacks.

The following table lists the modules, their assignments, their correctness assessments, where they are considered in the resistance ratings and the interfaces. Details on the vulnerability assessment that is concluded by the resistance rating are given later in the document.

Element	Assignment	Correctness in realizing SFRs	Resistance rating	Interfaces
Hardware	Enf./Sup.	YES	YES	TSFI: external user Physical & electrical
Crypto Lib	Enf./Sup.	YES	YES	internal
Flash Loader	not activated	YES	YES	none
Private algo	Non TSF	NO	NO	internal
Crypto ext. Communication Memory Mngt Security Services	Enf./Sup.	YES	YES	internal
Global Platform	Enf./Sup.	YES	YES	TSFI: external user & applis
GP non interf.	Non-interf.	NO (see note 1)	YES (see note 1)	TSFI: external user & applis (see note 1)
JCS	Enf./Sup.	YES	YES	TSFI: applis
Prop. APIs	Enf./Sup.	YES	YES	TSFI: applis
Suppl. APIs	Non TSF	NO	NO	not TSFI: applis
Specific Appli	Non TSF	NO	NO	not TSFI: external user
Framework1	Enf./Sup.	YES	YES	TSFI: applis
Framework2	Non TSF	NO	NO	not TSFI: applis
Security Domain1	Enf./Sup.	YES	YES	TSFI: applis
Application A	Enf./Sup.	YES	YES	TSFI: external user
Application B	not activated	NO	NO	none

Legend:

- Enf./Sup. = security enforcing or supporting element
- Non-interf.= security non-interfering element
- applis = applications

Note 1:

For GP non-interfering element the ADV assurance class has been assessed in terms of non-bypassabbility. The ADV evidences were available for vulnerability analysis. The resistance rating concerns the violations of SFRs that are not implemented by this element. The TSFI of this element does not map any SFR.

</EXAMPLES>

2.4. Description of TOE security mechanisms

<This chapter shall provide an overview of the implemented security mechanisms.>

This chapter shall support the understanding of the underlying platform and the implemented countermeasures by the composite evaluator. Data Sheets of hardware platforms as well as platforms including hardware and operating system include the description of flags or return codes that indicate errors. For the penetration testing it is important to get more information on the mechanisms triggering these flags or return codes to assess the composite TOE. If applicable, the "Description, remark" of the table shall include a reference to the Data Sheet or User Guidance Manual. If an ETR_COMP is a result of a composite evaluation itself security services of the underlying hardware platform that are relevant for the self-protection and non-bypassability of the TOE should be included too.

<EXAMPLE>

Security mechanism	Rely on	Description, remarks
SPA/DPA counter-measures	Feature	Clock generation, with countermeasures like jitter, cycle stealing. These mechanisms have to be activated by the embedded software.
Hardware AES	Design	Hardware AES co-processor. Cannot be changed because completely part of the glue logic.
Key loading AES	Software library	Key loading function that loads the key bytes in randomised order.
Secure RSA-CRT (recombination)	Software library and design	The TOE implements a secure RSA-CRT using the big number arithmetic co-processor provided by the certified IC. The RSA-CRT is implementing a so-called Montgomery exponentiation (square and always multiply) in which both the exponent and the message are blinded if enabled by the user. During the recombination phase the blinding is still in place.
Protection for Padding for RSA (PKCS#1)	Software	The padding verification of a message is performed in a time independent manner.
Secure compare	Software	The OS implements a secure compare function that

Security mechanism	Rely on	Description, remarks
function		can be used for signature verification purposes.
Memory protection	HW design	Hardware fault detection for memories.
Light sensors	HW design	Light sensors.
Randomised location of components	HW design	Components as CPU, co-processor and registers are put in randomised location, and covered by randomised wiring.

Table 2 – Architectural design

Legend:

- Security mechanism: title of the security mechanism
- Rely on: to be selected among “technology, feature, design, or software library”
- Description, remarks: description of the security mechanism, and the protection provided to counter threat or part of threat

</*EXAMPLE*>

3. Evaluated configuration

3.1. TOE configuration

<Describe the possible configurations of the product, and identify among them which ones have been covered by the evaluation, whereby this can include new configurations that are introduced as a result from an update that took place. Configurations of the TOE that are not certified must also be clearly identified, including a method of identification to allow verification by the user.>

<EXAMPLE>

The product can be in one of these possible configurations:

- Test configuration: TOE configuration at the end of developer IC manufacturing. The TOE is tested with a part of the Dedicated Software (called "XXX") within the secure developer premises. Pre-personalization data can be loaded in the EEPROM. The TOE configuration is changed to "*<intermediate>*" before delivery to the next user, and the part cannot be reversed to the "test" configuration.
- *<loader>* configuration: Depending on the product life cycle context chosen by the Embedded Software developer, the TSFI related to the Flash Loader may be accessible in Phase 3, Phase 4 or Phase 5 (in an exclusive way) or not accessible by the ES developer. This is seen as a specific TOE configuration that should be addressed in this example.
- *<intermediate>* configuration: TOE configuration when delivered to users involved in IC packaging and personalization. Limited tests are still possible with the Dedicated Software (System Rom operating system). Personalization data can be loaded in the EEPROM. The TOE configuration is changed to its final "User" configuration when delivered to the end user (the part cannot be reversed to the configuration).
- User configuration: Final TOE configuration. The developer test functionalities are unavailable. The Dedicated Software only provides the power-on reset sequence and routine libraries (mainly cryptographic services). After the power-on reset sequence, the TOE functionality is driven exclusively by the Embedded Software.
- New TOE end use configuration: Updated final TOE that is resulting from for example an update during usage phase of the TOE.
- *<Others such as I/O interfaces, memory sizes, additional libraries, protocols, ...>*

All configurations were evaluated (the last two configurations, i.e. "*<intermediate>*" and "User configuration", are those of the TOE in the user environment).

</EXAMPLE>

3.2. TOE identification method

<Describe the way to identify the microcontroller and its software libraries during composite evaluation. This has to be written in consistency with chapter 1.2.>

<EXAMPLE>

The following marks are physically printed (i.e. always visible) on the chip surface:

- IC identification: *<reference printed>*
- Dedicated software (*<test software, crypto libraries, other libraries>*) identification: *<reference printed>*
- Embedded software (in this case *<name of the software embedded for evaluation needs>*) identification: *<reference printed>*
- Manufacturing site identification: *<reference printed and meaning>*

Device identification can also be performed using *<specific register or memory content or command>*, which content *<or answer>* should be hexadecimal "0xXX" (see [AGD-X], section XXX).

Silicon revision can also be checked using *<specific register or memory content or command>*, which content *<or answer>* should be hexadecimal "0xYY" (see [AGD-X], section XXX).

Software library *<identification of the library>* can be checked using *<specific command>*, which answer should be hexadecimal 0xXXYY (see [AGD-X], section XXX).

<Repeat for all libraries or software parts.>

</EXAMPLE>

3.3. TOE installation, generation and start-up procedures

<If applicable for the platform, security relevant generation or installation parameter settings should be explained and their effects on the defence of attacks be outlined (e.g key length, counters limits).>

<EXAMPLE>

Installation/generation/start-up (IGS) operations are those needed to be performed by customers (i.e. users outside the developer's environment) to proceed the TOE (here: an IC) from the realization of its implementation (i.e. at the end of wafer fabrication) to its customer configuration (i.e. ready to be used: TOE in *<precise the different modes like "intermediate" and/or "user">* configurations).

For the specific case of a smart card IC, these operations correspond to those modifying the IC functionality and configuration. For instance:

- Personalization operations
- Configuration changes

For the <Name product> which was evaluated in "open mode" (i.e. without any specific embedded application), there is no personalization operation.

As for the "test" to *<"intermediate" or "user">* configuration change, it is performed only by the developer and is part of the developer manufacturing operations. After delivery the

TOE only features one fixed configuration (“user” mode), which cannot be altered by the user.

In conclusion, there is no customer preparative procedure, except for secure acceptance of the TOE.

</EXAMPLE>

4. Life-cycle

4.1. Introduction

The deliveries that are addressed in this chapter are the deliveries to external parties as identified in [ALC_DEL] and the life-cycle description in the security target of a given TOE.

For the composite evaluation of an OS on an IC the description of phase 1 and 4 are needed and will be detailed in this document. We should add also the delivery of the IC dedicated software and guidance to the application developer, and also identify the detail of fab-key protection mechanisms.

For an IC, as per the evaluation guide "The application of CC to Integrated Circuits" (cf. [JIL IC]), the deliveries under consideration are:

1. The delivery of the embedded application code to the microcontroller manufacturer, (in case of Flash products this may be replaced by the delivery of a key from the microcontroller manufacturer to the developer of the Security IC Embedded Software).
2. The delivery of the microcontroller to the entity in charge of the next step (testing, embedding into micro-module, card manufacturing).

For an OS, the deliveries under consideration are:

1. The delivery of the embedded application code to the manufacturer (if the code will be embedded in ROM) or product integrator (if the code will be embedded in EEPROM or Flash).
2. The delivery of the smart card/platform (IC with embedded OS) to the product integrator or personaliser or others in charge of the next step.
3. The delivery of the security guidance.
4. The exchange of key-material for access to the smart card/platform (IC with embedded OS).

4.2. Identification of the sites involved in the life-cycle model

The product life-cycle is the following:

<EXAMPLE>

Company	Address	Function/role in the life-cycle model	Site audit date
WWW		Libraries development	
XXX		IC design (development)	
AAA		Shipment of wafer	
YYY		Shipment of modules	
DDD		Provision of TOE documentation	

Table 3 – Identification of sites in the life-cycle model

</EXAMPLE>

All sites were evaluated. The environmental CC requirements (ALC) are fulfilled.

4.3. Deliveries between TOE manufacturer and embedded software developer

<Identification of the entry point, and description of the process for delivering any sensitive information (dedicated software, embedded software, data, documentation, tools, etc.).>

<Identification of any form, procedure [DEL], tools and process for integrity checks.>

<Identification of deliverables.>

4.4. Delivery from the TOE manufacturer to the card manufacturer

<Identification of the packaging of the product (wafer sawn or unsawn, module, etc.).>

<Identification of the entry point, and description of the process for delivering the IC and its documentation to the card manufacturer.>

<Identification of any form, procedure [DEL], tools and process for integrity checks (documentation, fab-key).>

<Identification of deliverables [AGD-X], IC, Fab-Key.>

4.5. Delivery from the embedded software developer to the product integrator

<Identification of the entry point, and description of the process for delivering any sensitive information (dedicated software, embedded software, data, documentation, tools, etc.).>

<Identification of any form, procedure [DEL], tools and process for integrity checks.>

<Identification of deliverables.>

4.6. Delivery from the TOE manufacturer to the product integrator

<Identification of the entry point, and description of the process for delivering any sensitive information (dedicated software, embedded software, data, documentation, tools, etc.).>

<Identification of any form, procedure [DEL], tools and process for integrity checks.>

<Identification of deliverables.>

5. Penetration testing

5.1. Introduction

The independent vulnerability analysis has been performed according to [CC] and [other methods required by the certification body]. The ratings have been calculated according to "Application of Attack Potential to Smartcards and Similar Devices" document (cf. [JIL AP]).

This chapter presents the list of attack scenarios that have been considered. The presentation of the different attack scenarios follows the examples given in [JIL AP].

The following descriptions should provide sufficient details to reproduce attacks which require countermeasures in the composite TOE.

To support the composite evaluator the evaluator of the platform shall include the results of his worst case analysis.

Each attack scenario shall follow the model/structure as depicted in the following section.

5.1.1. <Attack scenario – ID of attack scenario, e.g. AS-X, or DPA_AES…>

Attack step

<Method used shall be identified – effects obtained shall be described.>

<If sample preparation is done e.g. thinning of the substrate, it should be part of the attack step description.>

Date and history

<Date of the test performed. When the ETR is updated following surveillance period or re-evaluation, the history of testing activities shall be detailed: new analysis, evolution of the state of the art, new test or enhancement of test shall be detailed.>

CC parameters involved

CC parameters	Values
Security mechanism	
Security function/service	
SFR	
Objectives	
Assets*	

*For an IC or platform evaluation, the assets can be generic ones (as identified in the security target), e.g.: source code of possible embedded application, possible embedded application secret keys or confidential data loaded in memory, or services provided by the platform (RNG, firewall) that can be broken.

Test results

<Short description of the test that has been performed and relevant parameters, to provide information to the Application evaluation and to support the information on attack potential calculation.>

Information on attack potential

<Description of the attack, and discussion with details for each of the following parameters (cf. [JIL AP]):

1. *Elapsed time*
2. *Expertise*
3. *Knowledge of TOE*
4. *Access to the TOE*
5. *Open Samples/Samples with known Secrets*
6. *Equipment*

Rating

Factor	Ident'n	Exploit'n
Elapsed Time		
Expertise		
Knowledge of TOE		
Access to TOE		
Open Samples / Samples with known Secrets		
Equipment		
Sub Totals		
Totals		

Rationale

<If there is no rating, provide a justification. If the rating is over 31 provide it.

If the attack scenario is not feasible as far as some specific software countermeasure are applied, they shall be identified (e.g "see countermeasure XXX described in guidance [AGD-X], chapter Y.Z)".>

Test conclusion

<Conclusion on the penetration test, including the preconditions in the composite TOE that are required for the attack to be applicable (e.g. fixed key or message, the attacker needs to be able to load keys, etc.) and a reference to user guidance requirement if this is necessary to mitigate the risk of the attack.>

<In the case where a test performed on the platform indicates a possible attack path for which countermeasures must be implemented by the composite product, the technical information shall provide sufficient information for the composite evaluator to set up a similar attack path in order to validate the robustness of the countermeasures. This information shall include the general outline and idea of the attack and any technical detail specific to the TOE that proved to be important for performing the attack. Also included should be any observation from the testing activity that could highlight critical points for the composite evaluator.>

<EXAMPLES>

Note: The examples provided in the following sections 5.2, 5.3 and 5.4 only intend to illustrate the description of attack scenarios according to the model/structure outlined above.

The contents of these descriptions of attack scenarios cover different types of attack scenarios, but neither make claims of being complete nor being up to date.

5.2. Examples for Side Channel Attacks

5.2.1. Attack scenario – T.AES-Key-Load (Side Channel Attack)

Attack step
Using a template attack, the attacker aims at retrieving the key during the loading stage of the AES calculation.

Date and history
Initial test, performed <month year>

CC parameters involved

CC parameters	Values
Security mechanism	Random delays, clock-jitter
Security function/service	AES (AES-256)
SFR	FCS_COP.1/AES, FPR_UNO.1
Objectives	O.CIPHER
Assets	Secret key

Test Results
During a preliminary investigation it turned out that power signals show more distinct features and stronger peaks than the EM signal. Therefore, the attack was performed using power. With 40,000 training traces and 10,000 challenge traces it was possible to retrieve 248 out of the 256 key bit values. Based on this result the evaluator concluded that the TOE is not sufficiently protected against side channel attack on the key loading.

Information on attack potential

Factor	Ident'n	Exploit'n
Elapsed Time	< 1 week (2)	< 1 week (4)
Expertise	Expert (5)	Proficient (2)
Knowledge of TOE	Resticted (2)	Public (0)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	Public (0)	Public (0)
Equipment	Specialized (3)	Specialized (4)
Sub Totals	**12**	**10**
Totals	**22**	

Rationale
Measuring 50,000 and processing traces for the profiling phase takes a couple of days. This applies both for the identification and the profiling. For identification the attacker must be an expert for setting up the measurement and specifying the algorithms for further signal

processing. Only a proficient attacker is required when the attack is repeated in the exploitation phase. The usage of the command requests profiling. The attack is non-invasive and the equipment required to collect are a high-end digital oscilloscope, a probe and analysis software, which are considered specialized.

Conclusion
The TOE is not sufficiently protecting the key loading phase of the AES against side channel attacks when the blinding is disabled. When protection against side channel attacks is required the user shall use the key loading procedure as specified in section x.x of the user guidance. When the key bytes are loaded in random byte order the TOE is considered resistant to an attacker with high attack potential.

Especially for reassessments the evaluator shall also include the number of operations that ensure less than x broken bits (x according to the formulae within the attack methods paper). This means an attack is not applicable if the co-processor performs this number of operations using the same key.

5.2.2. Attack scenario – T.AES-Key-Operation (Side Channel Attack)

By collecting EM traces and performing a corresponding analysis, an attacker tries to recover secret keys during the AES operation.

Attack step
Using EMA analysis, an attacker is able to derive the key during the AES operation.

Date and history
Initial test, performed <month year>

CC parameters involved

CC parameters	Values
Security mechanism	Random delays, clock-jitter
Security function/service	AES
SFR	FCS_COP.1/AES, FPR_UNO.1
Objectives	O.CIPHER
Assets	Secret key

Test Results
The beginning of the AES operation is marked by a trigger (on the I/O-Line). The power consumption of the TOE during the AES operation is measured using a digital sampling oscilloscope. The emanation of the IC was measured with an EM probe adjusted near sensitive circuitry and analysed to extract the secret keys. Spatial analysis for EM probe acquisition is performed in order to determine optimal position for signal acquisition.

A post-processing is applied to each trace using filtering and elastic alignment based on the beginning of the operation that is used as reference value. The TOE is required to identify the attack path since the co-processor is exclusively implemented on this hardware platform.

A template attack is applied based on the aligned traces. The template is generated using one million traces. For the exploitation 500,000 traces are needed to reveal the key. The templates need a specific alignment due to chip individual differences. 150,000 AES operations using the same key can be applied without allowing more than 18 broken bits.

Information on attack potential

Factor	Ident'n	Exploit'n
Elapsed Time	< 1 month (3)	< 1 month (6)
Expertise	Expert (5)	Expert (4)
Knowledge of TOE	Restricted (2)	Public (0)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	Public (0)	Public (0)
Equipment	Specialized (3)	Specialized (4)
Sub Totals	**13**	**14**
Totals	**27**	

Rationale
Identifying the right location and collecting 1 million EM traces takes about one week. The signal analysis for alignment and performing the actual attack to retrieve the secret key bytes takes less than a month over all for identification. It is required to compress or stretch the templates depending on characterisation applied during the exploitation phase. Therefore the exploitation requires also more than a week and an expert is needed to adjust the templates. Since this is a hardware platform the results are generated using the test software of the evaluator using open samples. However, since it is a hardware platform evaluation open samples are not rated. The knowledge required to apply the attack is on Data Sheet level (restricted). The attack is non-invasive and the equipment for the collection of the traces is a high-end low noise digital oscilloscope with magnetic near field probe and analysis software is rated as specialized.

Conclusion
The key cannot be revealed if the AES coprocessor is limited to 150,000 operations using the same key. In case more than 150,000 operations need to be applied using the same key additional countermeasures must be implemented by the Security IC Embedded Software. These countermeasures must be analysed during the composite evaluation based on the specific IC Dedicated Software.

5.2.3. Attack scenario – T.RSACRT.RECOM (Side Channel Attack)

Attack step
Using EMA analysis, the attacker aims at extracting the private key from measured EMA traces during a signing operation with the RSA algorithm in CRT mode. The test aims at the recombination phase of the algorithm. During the test it is possible to sign many different messages and to recode the generated signature. Furthermore, the exponent blinding and message blinding is disable during testing.

Date and history
Initial test, performed <month year>

CC parameters involved

CC parameters	Values
Security mechanism	Random delays, clock-jitter

Security function/service	RSA-CRT
SFR	FCS_COP.1/RSA-SIGN, FPR_UNO.1
Objectives	O.CIPHER
Assets	Private key

Test Results
During a preliminary investigation it turned out that EMA signals show more distinct features and stronger peaks than using the power signal. Therefore, the test is performed on EMA. Using 1,000,000 traces it was possible to retrieve 20 bits of the private exponent dP. Based on this result the evaluator concluded that the TOE is not sufficiently protected against side channel attack on the RSA-CRT combination, when the blinding is disabled.

Information on attack potential

Factor	Ident'n	Exploit'n
Elapsed Time	< 1 month (3)	< 1 week (4)
Expertise	Expert (5)	Proficient (2)
Knowledge of TOE	Public (0)	Public (0)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	Public (0)	Public (0)
Equipment	Specialized (3)	Specialized (4)
Sub Totals	**11**	**10**
Totals	**21**	

Rationale
Identifying the right location and collection of 1 million EM traces takes several days. The signal analysis for alignment and performing the actual attack to retrieve the secret key bytes takes less than a month for identification and less than a week for exploitation. The attacker must be an expert to set up the measurement, and specifying the algorithms for signal analysis and retrieving the key bytes. Since this is a Global platform Java Card product neither open samples are needed nor specific knowledge for the TOE is required. The attack is non-invasive and the equipment to collect is a high-end digital oscilloscope, probe and analysis software is considered specialized.

Test Results with blinding enabled
The experiment was repeated with the blinding enabled as specified in the user guidance. With the blinding enabled it was not possible to recover bits of the targeted private exponent dP.

Conclusion
The TOE is not sufficiently protecting the RSA-CRT recombination phase against side channel attacks when the blinding is disabled. When protection against side channel attacks (DPA/DEMA) is required the user shall enable the blinding as specified in section x.x of the user guidance. When blinding is enabled the attack becomes not practical, and the TOE is considered resistant to an attacker with a high attack potential.

5.3. Examples for Fault Injection Attacks

5.3.1. Attack scenario – T.MMU (Fault Injection Attack)

Attack step
The aim of the attack is to access protected memory without authorisation and modify the user data, or read out the content of protected memory without authorisation. The attacker uses a Laser Fault Injection attack to disturb the MMU configuration in a way that the separation between different memory areas/two applications is circumvented.

If sample preparation is done like thinning, it should be part of the attack step description. In this example the IC form factor was considered appropriate to perform the attack.

Date and history
Initial test, performed <month year>.

CC parameters involved

CC parameters	Values
Security mechanism	Memory Management Unit
Security function/service	Access Control
SFR	FDP_ACC and FDP_ACF
Objectives	O.Memory Access Control
Assets	Integrity and confidentiality of user data

Test Results
Test software implementing access to dedicated memory areas with configured access rights was executed by the CPU. Based on the test results it was possible to disturb the hardware platform by laser fault injection on the back (silicon/substrate) side in a way that access to memory areas is provided although it shall be denied based on the configured access conditions. The attack is reproducible with a probability of about 20% to change the configuration of the MMU in the same way. The timing of the attack is relevant but not sophisticated. The length of the data read by the test software could not be changed during the tests. More effects could be seen on the back side than the front side. However, precise location of the vulnerable region showed to be critical.

Information on attack potential without implemented countermeasures as outlined in the user guidance

Factor	Ident'n	Exploit'n
Elapsed Time	< 1 month (3)	< 1 week (4)
Expertise	Expert (5)	Proficient (2)
Knowledge of TOE	Restricted (2)	Public (0)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	Public (0)	Public (0)
Equipment	Specialized (3)	Specialized (4)
Sub Totals	**13**	**10**
Totals	**23**	

Rationale

Identifying the right location to manipulate the configuration of the access rights (i.e. finding the register storing the corresponding access configuration data during operation of the device) and determine the reproducibility requires the scanning of the digital area of the chip and takes more than a week but less than a month. For exploitation the location is known, but due to the reproducibility rate and the time for interpreting the results the exploitation requires less than a week. To identify the spot and analyse all results an expert is required for identification. Exploitation can be done by a proficient level attacker. Since this is a hardware platform the results are generated using the test software of the evaluator using open samples. However, since it is a hardware platform evaluation open samples are not rated. The knowledge required to apply the attack is on Data Sheet level (restricted). The attack equipment for the collection to perform the perturbation is a laser setup rated as specialized.

Information on attack potential considering the user guidance

The user guidance requires that the integrity of the configuration data of the MMU is checked before reading or writing. The reference implementation example can be found in section x.x. This is added in the test software as an example and all disturbed accesses are detected. This is considered to be sufficient because the fulfilment of the user guidance is required by PP-0084.

Test Results

If the test software comprises countermeasures as described in the user guidance it was no longer possible to disturb the hardware platform by laser fault injection in a way that access to memory areas is provided although it shall be denied based on the configured access conditions. All attempts to change the MMU configuration were detected by the software countermeasures. Further experiments with multiple fault injection were discarded due to the usage of the reference implementation from the user guidance as secure verification function.

Factor	Ident'n	Exploit'n
Elapsed Time	*	*
Expertise	Expert (5)	Proficient (2)
Knowledge of TOE	Restricted (2)	Public (0)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	N/A	N/A
Equipment	Specialized (3)	Specialized (4)
Sub Totals	*	*
Totals	*	

Rationale

Identifying the right location to manipulate the configuration of the access rights and determine the reproducibility rate is performed without the countermeasures described in the guidance. It takes more than a week but less than a month. Then the attack is continued with the reference implementation. With single laser shots no successful attack could be mounted. Bypassing added countermeasure checks are considered not practical due to the usage of secure function verification reference implementation. Therefore the whole attack is rated as not practical.

Conclusion
Based on the analysis performed during the evaluation of the hardware platform the attack is considered to be not applicable on both sides of the IC if the countermeasures are implemented by the Security IC Embedded Software as required by the user guidance. Then it is mandatory to follow the directives presented in the user guidance document, section x.x.

5.3.2. Attack scenario – T.DFA_AES (Fault Injection Attack)

Attack step
Using a Laser Fault Injection attack the AES coprocessor can be disturbed in a way that the key can be compromised.

Date and history
Initial test, performed <month year>

CC parameters involved

CC parameters	Values
Security mechanism	Light Detection
Security function/service	AES
SFR	FCS_COP.1/AES
Objectives	O.Cipher
Assets	Secret key

Test Results
During AES encryption the co-processor can be disturbed with a low probability between 5% and 10% due to the light detection functionality. It was not possible to detect the timing for the fault attack, therefore faulty results must be grouped to support an attack. However, if a sufficiently high number of faulty crypto operations can be collected, it is possible to exploit the key. The attack is only applicable to fixed keys stored in the card. Session keys cannot be attacked because the session key is invalidated in case the fault attack is detected.

Information on attack potential without user guidance

Factor	Ident'n	Exploit'n
Elapsed Time	< 1 month (3)	< 1 month (6)
Expertise	Expert (5)	Expert (4)
Knowledge of TOE	Restricted (2)	Public (0)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	Public (0)	Public (0)
Equipment	Specialized (3)	Specialized (4)
Sub Totals	**13**	**14**
Totals	**27**	

Verification of the reference implementation
The user guidance requires a double calculation with a short random delay between both crypto operations. The random delay eliminates the possibility of double fault attacks with fixed delay. If the countermeasure is added all fault attacks were identified during the tests.

This is considered to be sufficient because the fulfilment of the user guidance is required by PP-0084.

Information on attack potential with user guidance

Factor	Ident'n	Exploit'n
Elapsed Time	*	*
Expertise	Expert (5)	Proficient (2)
Knowledge of TOE	Restricted (2)	Public (0)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	Public (0)	Public (0)
Equipment	Specialized (3)	Specialized (4)
Sub Totals	**13**	*
Totals	**not practical**	

Rationale
The vulnerability can only be identified if fault attacks are performed on open samples without countermeasures. Although it is possible to identify and verify the behaviour during the hardware evaluation (without countermeasures) it may be difficult to even find the weak location due to the response of the Security IC based on detected attacks. The countermeasures shall prevent further characterisation of the weakness even if this is not addressed.

Conclusion
Based on the analysis performed during the evaluation of the hardware platform the attack is considered to be not applicable if the countermeasures are implemented by the Security IC Embedded Software as required by the user guidance. However the composite evaluator must take care that the software is implemented according to the guidance.

5.3.3. Attack scenario – T.SIGNATURE-VERIFICATION (Fault Injection Attack)

Attack step
For signature verification the TOE implements a secure compare function, that is generally used for all signature verification functions that are supported by the TOE. The goal of the attack is to skip the secure compare function such that the TOE accepts an incorrect signature. The attack is performed on the AES-CMAC.

Date and history
Initial test, performed <month year>

CC parameters involved

CC parameters	Values
Security mechanism	From HW: clock-jitter, randomised location of components, light sensors, hardware fault detection for memories, randomised internal execution From SW: Random delays, redundancy
Security function/service	AES-CMAC signature verification

SFR	FCS_COP.1/AES-CMAC, FPR_UNO.1
Objectives	O.CIPHER
Assets	Secure compare mechanism

Test Results
The experiments show that it is possible to successfully skip the signature verification once every 100 manipulations. These manipulations are performed using green light at the metal side of the chip and infra-red light at the silicon side of the chip. Both sides of the chip show similar results. All successful attempts are performed in the random logic building block**.**

Information on attack potential

Factor	**Ident'n**	**Exploit'n**
Elapsed Time	> 1 month (5)	< 1 week (4)
Expertise	Expert (5)	Proficient (2)
Knowledge of TOE	Restricted (2)	Public (0)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	N/A	N/A
Equipment	Specialized (3)	Specialized (4)
Sub Totals	**15**	**10**
Totals	**25**	

Rationale
De-processing and initial testing to identify the failure combined with demonstrating the vulnerability takes more than 1 month to find the right combination of trigger and location (restricted command information is available). The exploitation will be based on a description of the attack and the commands to use, and therefore it will take less time. In order to find the right combination of trigger and location the attacker needs expert knowledge. He has to analyse the power traces and define the pattern recognition. A proficient rating is required for exploitation because of the de-processing techniques and equipment operation required. The attack requires restricted information to identify exploitable parts of commands, but the command is scripted for the exploitation phase (hence public). More than one sample may be necessary but less than ten. Regarding effort spend in identification, no open samples are necessary. Minimum equipment is used to de-process the chip, to bond out the pads and to generate and analyse the required commands to run the IC. A laser and optical microscope are required to generate the perturbation and a digital scope is used to identify and repeat the attack timing.

Conclusion
The experiments show that an attacker is able to manipulate the secure compare function and skip the signature verification. A user of the function has to implement additional countermeasures as outlined in section x.x. of the guidance or limit the attack window as outlined in section x.x of the guidance.

5.3.4. Attack scenario – T.Integrity_protection_Memories (Fault Injection Attack)

Attack step

The TOE implements integrity protection of the user data stored in the memories. The aim of the attack is to manipulate the TOE such that the memory is manipulated, and the integrity mechanism is circumvented. Both the RAM and the EEPROM are attacked.

Date and history

Initial test, performed <month year>

CC parameters involved

CC parameters	Values
Security mechanism	clock-jitter, randomised location of components, light sensors, hardware fault detection for memories
Security function/service	Integrity protection
SFR	FDP_SDI.2
Objectives	O.Phys-Manipulation
Assets	User data stored in memories

Test Results

The experiments show that it is possible to successfully manipulate the integrity mechanism once every 1000 manipulations attempts. The manipulations are performed using green light at the metal side of the chip and infra-red light at the silicon side of the chip. All successful attempts are performed in RAM (address logic) and the attacks at the silicon side of the chip were more effective. Successful attacks were exercised when the RAM was written. The attack was not successful on EEPROM.

Information on attack potential (for RAM)

Factor	Ident'n	Exploit'n
Elapsed Time	< 1 month (3)	< 1 week (4)
Expertise	Expert (5)	Expert (4)
Knowledge of TOE	Restricted (2)	Public (0)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	Sensitive (4)	Public (0)
Equipment	Specialized (3)	Specialized (4)
Sub Totals	**17**	**12**
Totals	**29**	

Rationale

It takes less than a month to prepare few samples in the same way as required for fault injection, record traces and analyse the gathered information, then identify the sensitive areas and next repeat to estimate the repeatability of the attack. In order to find the right combination of trigger and location the attacker needs expert knowledge to build the test bench for performing the attack. He has to analyse the power traces and define the pattern recognition. Expert level is necessary to repeat the attack, taking into account the difficulties caused by the implemented countermeasures. The attack requires restricted information to

identify the exploitable commands, but the command is assumed to be scripted for the exploitation phase. Open samples are managed accordingly. Minimum equipment is used to de-process the chip, to bond out the pads and to generate and analyse the required commands to run the IC. A laser and an optical microscope are required to generate the perturbation and a digital scope is used to identify and repeat the attack timing.

Conclusion
The experiments show that an attacker is able to manipulate the integrity mechanism implemented in RAM. A user of the function has to implement additional countermeasures as outlined in section x.x. of the guidance. The experiments also show that the integrity mechanism is sufficiently protecting EEPROM and therefore rating becomes not practical.

5.4. Examples for Software Attacks

5.4.1. Attack scenario – T.Bleichenbacher (Software Attack)

Attack step
The TOE implements a time-independent padding verification and provides a detailed error message informing the user whether the padding was correct or not. Based on this detailed error message an attacker can use the signature verification as an oracle and decrypt the ciphered message C.

Date and history
Initial test, performed <month year>

CC parameters involved

CC parameters	Values
Security mechanism	Time independent padding verification of messages
Security function/service	Signature verification
SFR	FCS_COP.1\RSA_PKCS#1
Objectives	O.Crypto
Assets	Decrypt a padded ciphered message

Test Results
The test experiments show that an attacker will not be able to retrieve any information on the timing or through side channel that allows distinction on whether or not the padding of the message was correct. However, he will be able to decrypt a ciphered message padded according to PKCS#1 using the chosen cipher text attack from Bleichenbacher using the detailed error messages.

Information on attack potential

Factor	Ident'n	Exploit'n
Elapsed Time	< week (2)	< 1 day (3)
Expertise	Proficient (2)	Proficient (2)
Knowledge of TOE	Restricted (2)	Public (2)
Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with	N/A	N/A

known Secrets		
Equipment	Standard (1)	Standard (2)
Sub Totals	**7**	**9**
Totals	**16**	

Rationale
An attacker with proficient knowledge will be able to identify the Bleichenbacher attack in a few days (less than a week). Once identified the exploitation will take less than a day. The expertise is still proficient, each result will require some work for the next step. The knowledge of the TOE is considered restricted during identification and public once the attack scenario is identified. No special equipment or prepared samples are needed for the attack, the attack is non-invasive and therefore the amount of samples needed for the attack scenario is limited.

Conclusion
The embedded software using the signature verification should follow up on user recommendation x.x making sure that only unspecified error messages are provided to the user. If the user guidance is followed up the attack becomes not applicable.

5.4.2. Attack scenario – T.Malicious applet (Software Attack)

Attack step
The TOE implements a partially defensive virtual machine. To test the strength of the implemented countermeasures, the evaluator has loaded a set of malicious applets containing well-known type confusion and illegal byte codes. The applets do not pass the byte code verifier.

Date and history
Initial test, performed <month year>

CC parameters involved

CC parameters	Values
Security mechanism	Partially defensive virtual machine
Security function/service	Firewall
SFR	FDP_ACC.2/Firewall, FDP_ACF.1/Firewall
Objectives	O.Firewall
Assets	Access sensitive data beyond the boundary of the current applets memory area

Test Results
The test experiments show that it is possible to create type confusion and exceed the working area of the applet.

Information on attack potential

Factor	Ident'n	Exploit'n
Elapsed Time	< 1 month (3)	< 1 week (4)
Expertise	Multiple expert (7)	Proficient (2)
Knowledge of TOE	Public (0)	Public (0)

Access to TOE	<10 (0)	<10 (0)
Open Samples / Samples with known Secrets	Restricted (2)	N/A
Equipment	Specialized (3)	Specialized (4)
Sub Totals	**15**	**10**
Totals	**25**	

Rationale
The identification of the attack takes less than a month but more than a week, a successful perturbation attack must be performed to load the applet and the data retrieved has to be interpreted. Once identified re-doing the attack during the exploitation phase takes less than a week. A multiple expert knowledge is necessary for performing the initial perturbation attack to load the applet and Java Card knowledge to interpret the data. Once the path is identified part of the work can be scripted, therefore proficient knowledge is enough during the exploitation phase. Only public knowledge and several samples may be needed for identification as well as for exploitation. For the interpretation of data retrieved from the TOE open samples might be needed. The equipment is specialized to perform a laser perturbation. The Java Card tools required are basic equipment.

Conclusion
The virtual machine is not fully defensive against malicious applets containing type confusion and illegal byte codes. However, the security requirements mandate that all applets that are loaded on the TOE pass the byte code verifier. Furthermore, the Global Platform authentication implementation is resistant against attackers with high attack potential. The full attack path, consisting of manipulating the authentication followed by loading a malicious applet is considered not practical.

</EXAMPLES>

5.5. <Iteration of attack scenarios>

<Describe all attacks following for each the model/structure given in the previous sections.>

[For list of attacks, refer to the last version of "Attack Methods for Smartcards and Similar Devices" (cf. [JIL AM]). This list shall be considered as a minimum.]

5.6. Summary

<Provide a table, listing vulnerabilities, associated attack scenario, description in what way the assurance was gained and assets involved, with a status, each attack method as identified in [JIL-AM] shall be addressed.>

<EXAMPLE>

The following table sums up penetration tests that have been performed, and their results:

Vulnerabilities	Attack scenarios	Assets involved	Assurance in protection	Status	Guidance
Physical Attacks					
Reading the	AS-03, *<or*	Content of ROM		OK	

Vulnerabilities	Attack scenarios	Assets involved	Assurance in protection	Status	Guidance
content of the ROM	*READ_ROM, or…>*	(embedded software)			
Physical obvservation	AS-07, *<or REVERS, or…>*	IC design		OK	
	Overcoming sensors and filters				
N/A	N/A	_	For assurance on overcoming sensors and filters, the TOE relies on the underlying IC certification including [ETR_COMP_IC].	OK	
…					
	Perturbation Attacks				
EEPROM perturbation	READ_EE, *<or AS-04, or…>*	Confidentiality of data in EEPROM	For assurance of data stored in the EEPROM, the TOE relies on the underlying IC certification including [ETR_COMP_IC] and the security recommendations in the guidance are followed up.	OK	
	T.integrity protection of memory	Integrity of data in EEPROM	The testing experiments show, combined with the design review that the IC is providing sufficient integrity protection for data stored in EEPROM.	OK	See [AGD-X], §X.Z
RAM perturbation	T.integrity protection of memory	Integrity of data in RAM	The testing shows that the IC is not sufficiently protecting the integrity of data in RAM and therefore the user has to implement additional countermeasures in the embedded software.	OK-S	See [AGD-X], §X.Z
	T.Signature verification	Secure compare function	The testing shows that the secure compare can be manipulated for all	OK-S	See [AGD], section x.x

Vulnerabilities	Attack scenarios	Assets involved	Assurance in protection	Status	Guidance
			signature verification algorithms. The user has to implement additional countermeasures in the embedded software.		
…					
…					
Retrieving keys with DFA					
…					
…					
Side channel attacks – Non-invasive retrieving of secret data					
Leakage information	SPA/DPA_AES, <*or…*>	Any key involved in AES calculation	Testing shows that the IC is not sufficiently resistant against template attacks and that the provided software library must be used to provide sufficient protection.	OK-S	See [AGD-X], §A.B
	SPA/DPA_RSA, <*or…*>	Any key involved in RSA calculation		OK-S	See [AGD-X], §Y.Z
	DPA RSA-CRT recombination	Any private key involved in the RSA-CRT calculation	Rely on the IC certification for the arithmetic co-processor (refer to platform ETR_COMP), penetration testing is performed to gain assurance for the RSA-CRT implementation.	OK-S	See [AGD], section x.x
…					
…					
Exploitation of Test features					
…					
…					
Attacks on RNG					
…					
…					
Ill-formed Java Card applications					
Firewall to separate applets	Execution of malicious applets	Data of different users, CAP files of different users	Rely on the Global Platform authentication to	OK-S	See [AGD_ADMIN],

Vulnerabilities	Attack scenarios	Assets involved	Assurance in protection	Status	Guidance
			protect against loading of malicious applets and the administrative guidance. Testing shows that malicious applets can create type confusion.		section x.x
Software Attacks					
N/A	Bleichenbacher attack	Secure message m	Assurance was gained through testing, the TOE is resistant against an attacker with high attack potential when the embedded software adheres to the user guidance.	OK-S	See [AGD-X], section x.x
<Others>					
…					
…					

Table 4 – Penetration tests

Legend:

- OK: OK without any additional countermeasures
- OK-S: OK with additional software countermeasures / OK when respecting the platform user guidance (gives precise reference to the guidance)

</EXAMPLE>

6. Assessment of supporting functions

This is a placeholder to add details on testing that is performed on for instance the arithmetic co-processor or the CPU, that is not modelled by SFRs but properties of these components are claimed in the Security Architecture description (ARC) and/or the user guidance (AGD_OPE).

7. Observations and recommendations

<The goal of this chapter is not to repeat the guidance recommendations, but to outline sensitive aspects that should be analysed carefully.

Provide any additional required information for a secure usage, or any additional information required for composite evaluation (cf. [COMP]).

Observations should accelerate the composite evaluation by supporting the separation between possible vulnerabilities and effects that do not allow to attack the platform.

Recommendations should sensitise the composite evaluator regarding configuration of the platform or possible combination of countermeasures that may lead to residual vulnerabilities that may be exploitable depending on the use case and the composite components.

Observations and recommendations represent the - to some extent subjective - results of the evaluator of the platform. They shall support the composite evaluator but they are not intended to limit the penetration testing scope of the composite evaluator.>

7.1. Observation

Observations are issues or remarkable behaviour identified by the evaluator during penetration testing of the TOE. Observations are not considered as residual vulnerabilities based on the assessment of the evaluator of the hardware platform. The description shall support the composite evaluator during test preparation and testing to prevent detailed analysis of issues or remarkable behaviour that is not considered as residual vulnerability. Since the observations are based on penetration testing they may not be obviously described in the Data Sheet and may be hard to assess without further design knowledge.

<EXAMPLE 1>

Evaluation result: The clock frequency of the hardware platform may differ depending on the call of routines provided by the IC Dedicated Support Software. This may be identified by the evaluator of the TOE e.g. during side channel analysis or fault injection testing.

If the TOE is a hardware platform with IC Dedicated Support Software the interfaces between the Security IC Embedded Software and the IC Dedicated Support Software are described. This shall include the hardware registers changed by the IC Dedicated Support Software. The IC Dedicated Support Software may or may not change any configuration (clock setting, countermeasures) of the hardware platform or not. The dependency is described in a subordinate clause of the guidance for the TOE that is sufficient for the developer of the Security IC Embedded Software. The evaluator of the TOE shall include an observation with the related reference to the user guidance, the impact on the configuration of the hardware platform and the assessment. Thereby the composite evaluator does not need to search for related information. Further on, it is clear to the composite evaluators that the evaluator of the hardware platform had considered this aspect and the results can be used during the composite evaluation.

</EXAMPLE 1>

<EXAMPLE 2>

Evaluation result: The output of a test routine provides unexpected answers during fault injection testing at a specific location. This may be identified by the evaluator e.g. during fault injection testing.

If the TOE is a hardware platform, the UART of the hardware platform may be sensitive to fault attacks. This may not be considered as exploitable by the evaluator of the hardware because the effect can be assigned to the UART and the same faults can be generated by disturbance of the communication. The evaluator can assess this effect based on the knowledge of the test software and design information of the hardware platform. This shall be mentioned as observation because the composite evaluator may also identify this sensitive spot but the composite evaluator may not be able to assign it to the UART because the design information is not available during the composite evaluation. If the information is not provided by the evaluator of the hardware to the composite evaluator, the composite evaluation requires additional testing and analysis to assess the identified effect.

</EXAMPLE 2>

<EXAMPLE 3>

Further examples shall be added, especially for composite evaluation of applets.

</EXAMPLE 3>

7.2. Recommendation

Recommendations are hints for the composite evaluator regarding the analysis of the composite product, the planning of tests and the checks regarding the specific preparation, configuration and start-up of the evaluated platform. This may include hints on specific configurations that can be selected by the customer or behaviour that depends on the components that may be added by the composition. A recommendation may indicate residual vulnerabilities depending on the usage and configuration of the platform.

<EXAMPLE 1>

Evaluation result: Successively adapted test software shows a significant impact on the timing behaviour of the test depending on the size of the test software. This may be identified by the evaluator of the TOE e.g. during side channel analysis.

The TOE may include caches to speed up the access to code and/or data during the execution. Based on the implementation and size of the cache this may have an impact on the behaviour of timing-invariant code. The user guidance includes sufficient information for the developer of the software to consider this issue. However, the evaluator of the hardware platform shall address this behaviour and add references to the related description in the user documentation with possible dependencies to support the orientation of the composite evaluator. Thereby it is clear that this aspect must be considered during the composite evaluation.

</EXAMPLE 1>

<EXAMPLE 2>

Further examples shall be added, especially for composite evaluation of applets.

</EXAMPLE 2>

Annex 1. References about the evaluated product

[AGD-1]	
[AGD-2]	
[AGD-ADMIN]	
…	
[CERTIF]	
[CONF]	
[DEL]	
[ETR]	
[ST]	
[ST-Lite]	
[ETR_COMP_IC]	
[HW_AGD]	

Annex 2. Methods and standards for certification

<National regulation applicable for IT certification>	
[CC]	CCMB-2022-11-001: Common Criteria for Information Technology Security Evaluation, CC:2022, Part 1: Introduction and general model, Revision 1, November 2022 CCMB-2022-11-002: Common Criteria for Information Technology Security Evaluation, CC:2022, Part 2: Security functional components, Revision 1, November 2022 CCMB-2022-11-003: Common Criteria for Information Technology Security Evaluation, CC:2022, Part 3: Security assurance components, Revision 1, November 2022 CCMB-2022-11-004: Common Criteria for Information Technology Security Evaluation, CC:2022, Part 4: Framework for the specification of evaluation methods and activities, Revision 1, November 2022 CCMB-2022-11-005: Common Criteria for Information Technology Security Evaluation, CC:2022, Part 5: Pre-defined packages of security requirements, Revision 1, November 2022
[CEM]	CCMB-2022-11-006: Common Methodology for Information Technology Security Evaluation, CEM:2022, Evaluation Methodology, Revision 1, November 2022
[COMP]	Joint Interpretation Library – Composite product evaluation and certification, *<version TBD (latest approved version)>*
[JIL AP]	Joint Interpretation Library – Application of Attack Potential to Smartcards and Similar Devices, *<version TBD (latest approved version)>*
[JIL AM]	Joint Interpretation Library – Attack Methods for Smartcards and Similar Devices, *<version TBD (latest approved version)>* (confidential document)
[JIL IC]	Joint Interpretation Library – The Application of CC to Integrated Circuits, *<version TBD (latest approved version)>*