Joint Interpretation Library

Security Architecture requirements (ADV_ARC) for smart cards and similar devices

Document purpose: provide requirements to developers and guidance to evaluators to fulfill the Security Architecture requirements of CC V3 ADV_ARC family.

Version 2.0
January 2012

This page is intentionally left blank

Table of contents

Contents

1 Introduction

1.1 Objective of the document

The current document provides requirements for the developer and guidance for the evaluator on how to apply the assurance requirements of the family ADV_ARC to the Technical Domain of smart cards & similar devices. The developer documentation provided to fulfil the ADV_ARC family is denoted as “ARC document” in the text.

The smart card technology requires special interpretation because it combines security integrated circuits, operating systems and applications to high secure devices. Therefore, this document is intended to provide mandatory interpretation for the application of the ADV_ARC family. It is addressed to both developers of security integrated circuits and developers of composite products, consisting of a hardware platform and embedded software. The embedded software can be organised in different ways (native software, closed operating systems with one or more applications, open software platforms and more).

The expected assurance level is EAL4 (augmented by at least AVA_VAN.5) or higher.

The mandatory interpretation defines what kind of information the ARC document SHALL contain and in which level of detail this information SHALL be provided. It does *not* define mandatory tasks for the evaluator. However, the document also serves as a guideline for the evaluator: in order to have a clear agreement between evaluator and developer, it states which kind of developer information is mandatory and may also define which is *not*.

An informative part is provided in the Appendix which contains examples for the type of information and level of detail to be provided in the ARC document.

1.2 Scope of Security Architecture

The version 3 of Common Criteria (CC) introduces a new security assurance requirements (SAR) family Security Architecture (ADV_ARC). Its objective is described in paragraph 214 of CC part 3 as follows:

> *“The objective of this family is for the developer to provide a description of the security architecture of the TSF. This will allow analysis of the information that, when coupled with the other evidence presented for the TSF, will confirm the TSF achieves the desired properties. The security architecture descriptions support the implicit claim that security analysis of the TOE can be achieved by examining the TSF; without a sound architecture, the entire TOE functionality would have to be examined.”*

A security architecture is a set of properties that the TSF exhibits; these properties include self-protection, domain separation, and non-bypassability. These properties are distinct from security functionality expressed by Part 2 SFRs because they largely have no directly observable interface at the TSF. Rather, they are properties of the TSF that are achieved through the design of the TOE and TSF, and enforced by the correct implementation of that design.

The Security Architecture shall also describe the TOE security functionality (TSF) initialisation, i.e. the processing that occurs in transitioning from the “down” state to the initial secure state, when power-on or a reset is applied.

Some features of the security architecture in CC version 3 were described as security functional requirements in CC version 2: non-bypassability was described by the SFR family Reference mediation (FPT_RVM) and domain separation by the SFR family Domain separation (FPT_SEP). When appropriate components from the families FPT_SEP and FPT_RVM were combined with the appropriate components from TSF internals (ADV_INT), the TOE can be said to have what has been traditionally called a "Reference Monitor" (cf. CC version 2.3, part 2, chapter 6 and annex J). As the families FPT_SEP and FPT_RVM are removed from CC part 2, the related security features shall now be described through the ADV_ARC family.

The Technical Domain of Smart card & Similar Devices presents specificities that have to be taken into account in the drawing up of the ADV_ARC documentation. The main characteristics are:

- A device belonging to this domain is a combination of a one-chip Integrated Circuit with embedded software implementing cryptographic services using secrets. The TOE could cover the full product or only a layer that includes the IC (an underlying platform).

- The TOE may start up in a low-function mode and then transition to the evaluated secure configuration. A transition from power off also happens each time the device is used by the final holder.

- In its operational environment an attacker might have physical access to the TOE through the physical port and the IC surfaces.

- The lifecycle is similar to that described in the "Guidance for smartcard evaluation" document.

Moreover the document focuses on devices that must be resistant to attackers with a high attack potential.

A number of supporting documents have been issued for this Technical Domain with which the current document is in coherency. In particular:

- "Application of attack potential to smart cards" and its companion document JIL – Attack methods for smart cards - provide guidance metrics to calculate attack potential required by an attacker to effect an attack following a list of state of the art attack method.

- "Composite product evaluation" defines the precise conditions for the re-use of the results from the evaluation of an underlying platform.

- "Smart card evaluation" defines smart card evaluation and certification terminology and describes appropriate advice.

2 General Aspects of Content and Presentation

ARC documentation supports the vulnerability analysis of the evaluator but it does not provide a developer vulnerability analysis. The developer designs, implements and describes the security architecture of the TOE. The ARC documentation describes security domains and the secure initialisation process; and demonstrates self-protection and non-bypassability. The description focuses on the use of security mechanisms which are put into place and their collaboration in order to achieve overall security. To this end the developer may analyse and conclude how the security features and countermeasures of the TOE are **intended** to resist the general attacks listed in the document “Application of Attack Potential to Smart Cards” viewed in the light of tampering and bypass. In contrast to the ARC document the evaluator performs an independent vulnerability analysis to determine the **actual** resistance of TOE to attacks. The evaluator shall consider all potential vulnerabilities encountered while performing evaluator activities or found by independent methodical search. The evaluator will determine whether vulnerabilities are exploitable by an attacker possessing the attack potential addressed in the ST. Thus ARC documentation and vulnerability analysis are different in responsibility, methods and result.

The security architecture description shall describe all properties of the TOE and the TSF and all security mechanisms of the TSF that contribute to enforce the security architecture. The security mechanisms specific for enforcement of security architecture properties maybe fully described

- in the ARC documents or
- in the TDS documentation and the ARC documentation refers to these descriptions.

Note some security mechanisms are spread across the whole implementation and cannot be expressed or are not easily expressible within TDS documents and mapping to modules. The description of the security architecture should avoid redundancy with other parts of ADV.

The CC requires the security architecture description being at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document. But this does not imply the same rigor of the presentation in the ARC documents; the use of semi-formal or formal methods is not required for ARC documents. Even though the CC requires the developer to provide a mapping between the TOE design description and the sample of the implementation representation, such mapping is not required for the security architecture description.

Within the Technical Domain of smartcards and similar devices the TOE physical boundaries are TSFIs. The device surface is the TSFI for physical protection against manipulation. The surface of the IC itself can output physical signals such as electromagnetic emanations that could be used for side channel analysis or input energy used for perturbation like laser attacks. The ports are physical entry or exit points of power supply and physical signals for the TOE that provides access to the TSF. The physical signal contains more information (e.g. timing, signal level) than the data intended to be exchange through the logically defined TSFI. The power supply port is not part of the logical interface but may affect the TSF (e.g. by glitches).

The evaluator is reminded that it is the synergy and not the distinction of self-protection, non-bypassability, domain separation and secure initialisation that are in the focus of the ARC documentation.

3 Level of description in ADV_ARC

ADV_ARC.1.1C requires the architecture description to be “at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document”.

As the expected assurance level is EAL4+ (EAL augmented by at least AVA_VAN.5) or higher, the level of description corresponds to parameters, actions and error message for TSFI, the module interface level and in some case to implementation specific details. But semi-formal or formal description is not required because it does not bring more comprehensive details.

The security architecture description is based upon security mechanisms (SFR-enforcing entities, mechanisms enforcing the properties, design countermeasures, coding conventions). Each security mechanism must be explained in terms of purpose and behaviour with the exception of SFR-enforcing entities that are described in decomposition documentation.

For Security Mechanisms spread across the whole implementation, it shall be ensured that there is little ambiguity between the description in ADV_ARC and ADV_IMP by providing the principles that have led to their implementation in the code. The security mechanisms description may be illustrated with code sample or example.

4 Security domains

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

Domain separation is a property whereby the TSF creates separate *security domains* on its own and for each untrusted active entity to operate on its resources, and then keeps those domains separated from one another so that no entity can run in the domain of any other.

The security architecture description explains the different kinds of domains that are created by the TSF, how they are defined in terms of resources allocated to each domain, and how the domains are kept separated so that active entities in one domain cannot tamper with resources in another domain.

If the TSF is the only active entity and there are only data structures maintained by the TSF to manage the interactions with the users, the security architecture will describe that there is no security domain available for active entities.

If the TSF provides security domains for other active entities the TSF shall protect their own domain against adverse actions of these potentially-harmful entities on TSF resources. Moreover, the TSF keeps this domain separated from the security domain of other active entities.

If the ARC documentation describes security domains the allocation and deallocation of the resources for the active entities should be under SFR control (e.g. FDP_ACC: access control). The use of the resources by the active entity in the security domain is outside TSF control. The active entities may use these resources according to their own security policies but they are not allowed usage of other resources outside their security domain. Therefore the domain description provided in the ARC documentation shall meet TSF access control to the security domain resources as expressed by the SFR and the other SFR must not contradict the security domain definition. If the ARC documentation describes security domains in term of resources not controlled by a SFR, that would mean that an SFR is missing.

In case of composite evaluation the applicative layer could rely upon the underlying platform to correctly instantiate the domains that the TOE defines. The developer should list the used security services offered by the platform to support security domain separation and make reference to these services in the description.

5 Secure start-up

The security architecture description shall describe how the TSF initialization process is secure (cf. ADV_ARC.1.3C). The information provided in the security architecture description relating to TSF initialisation is directed at the process bringing the TSF from the “down” state (e.g. power-off or after reset) into an initial secure state (i.e. when all parts of the TSF are operational, cf. CEM paragraph 530). For smart cards and similar devices

- parts of the TSF may be active even in power off e.g. physical protection against undetected manipulation,

- parts of the TSF may be temporally deactivated e.g. in power save modes.

The goal of the secure initialisation process of smart cards and similar devices is to enforce the security objectives even while some TSF parts are not active (i.e. during power off or power save modes) or in activation process (e.g. start-up) or in deactivation process (e.g. transition into power save mode). The secure initialisation process requires that self-protection and non-bypassability is ensured during these transitions. This implies that in any point of time the TOE function is not available if the TSF parts protecting this function are not activated.

The secure initialisation process will be implemented by specific security features or security functionality not directly following from SFR. This specific security functionality and their security mechanisms may be not described in other ADV assurance families. The objective of the ARC documentation for secure initialisation is to provide all the information required to treat these components as part of the TSF.

The secure initialisation process may implement mechanisms protecting the confidentiality or checking the integrity of the implementation of other TSF. Some mechanisms may be not needed after secure initialisation and shall be protected against misuse.

If external interfaces of the initialisation process are fully described as TSFI in terms of actions in ADV_FSP.4 and beyond or the mechanisms as part of the TSF are described in terms of purpose and interactions of modules in ADV_TDS.3 and beyond they do not have to be described again.

6 Self-protection

The component ADV_ARC.1.4C requires that the security architecture description demonstrates that the TSF protects itself from tampering.

Self-protection refers to the ability of the TSF to protect itself from manipulation from external entities that may result in changes to the TSF, so that it no longer fulfils the security objectives or SFRs.

Tampering with the TSF may be realized by untrusted active entity running on behalf of an external entity. Mechanisms that provide domain separation to define a TSF domain that is protected from other (user) domains would be identified and described.

Within the Technical Domain of SC&SD the TOE physical boundaries from which an external entity may intervene are the ports and the surface of the IC. The ports are physical entries of the TOE supporting logical interface that provide access to the TSF for physical parasitic signals. The surface of the chip may be also an entry point for physical parasitic signals. These signals may induce a modification of the stored code & data or of the correct execution of the code.

The functional requirement class FPT (Protection of TSF) contains families of functional requirements that relate to the integrity and management of the mechanisms that constitute the TSF and to the integrity of TSF data. Components from the class FPT are necessary to provide requirements that the SFPs in the TOE cannot be tampered.

Self-protection can therefore not generally be achieved by a mere implementation of an SFR but other security mechanisms may be added and collaborate with the security mechanisms implementing the SFR.

Self-protection of the TSF will be achieved by:

- Security mechanisms: the ability of each security mechanism to contribute to the protection against direct attacks.
- Binding of security mechanisms: the ability of the security mechanisms to work together in a way that is mutually supportive and provides an integrated and effective whole.
- Combination of hardware and software security mechanisms

The initialisation process shall guarantee that the TSF is in an initial secure state and had not been spoofed by any means. The developer shall explain how the initialization process checks the TSF code integrity. The integrity of the initialisation process code shall also be checked during this process.

In some cases the TOE starts up in a low-function mode, a mode whereby untrusted users are able to login and use the services and resources of the TOE. In this mode the code does not run in the evaluated configuration and these services are no more accessible.

In this case the security architecture description shall include an explanation of how the TSF is protected against this code in the evaluated configuration:

- what prevents this code from running
- what prevents those services from being accessible

In case of composite evaluation the platform could provide security services that contribute to the self-protection in cooperation with the application layer security mechanisms. The developer shall list the used security services offered by the platform and make reference to them in the following analysis.

The developer shall describe the security mechanisms and their collaboration to protect the TSF from tampering. The developer shall provide a description on how the TOE reacts in presence of the relevant attacks listed in the document “Application of Attack Potential to Smart Cards” and provide a conclusion.

7 Non-bypassability

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

Non-bypassability is a property that the security functionality as specified by the SFRs is always invoked and cannot be circumvented when appropriate for that specific mechanism (cf. to Annex A of CC part 3, paragraph 519).

7.1 TSF always invoked

Non-bypassability means firstly that there is no possibility to bypass the SFR-enforcing entity by using unexpected and undocumented paths in the design. Any possibility to bypass the TSF is therefore attributed to a flaw in the design or implementation.

From EAL4 level the functional specification shall describe all actions associated with each TSFI (ADV_FSP.4.4C) and the design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules (ADV_TDS.3.9C). In this case all modes or operations of TSFI are documented at a sufficient level to provide evidence of non-bypassability by exploiting a flaw in the design.

Secondly, non-bypassability requires that no Functional Interface can be used to violate the TOE security objectives, to circumvent SFR or to conflict with SFR. When Functional Interfaces exist the developer shall list them and explain either why they have no interaction with the TSF or why they are not providing a path for circumventing the TSF. In this case domain separation description (see the corresponding chapter) may bring evidence of non-bypassability.

Thirdly, non-bypassability deals with those cases where the attacker has only logical access to the TOE as opposed to the case of “tampering” which is to be countered by self-protection (see the corresponding chapter).

The developer shall describe the security mechanisms and their collaboration to protect the TSF from software attacks exploiting an insufficient design or implementation to meet the TOE security objectives. The developer shall provide a description on how the TOE reacts in presence of the relevant attacks listed in the document “Application of Attack Potential to Smart Cards” and provide a conclusion.

7.2 Side channel

Side channels are unenforced signalling channels carrying information about internal secrets, states or processes provided by monitoring of the processing of any object containing or related to this information (cf. CEM paragraph 1909). The information may be contained in any observable physical value as power consumption of the device, voltage and timing on ports of the output interfaces, electromagnetic emanation on IC surface. The signals of output ports may contain more information than the data intended to be exchange through the logical interface defined in the TSF documentation. The power supply interface and the electromagnetic emanation through the IC surface are not intended for information output at all but may carrying information.

The side channels bypass the TSF because they leak any information intended to be kept secret. The secret information include but are not limited to authentication reference data (e.g. for PIN verification), symmetric secret or asymmetric private cryptographic keys, timing of data processing enabling other attacks.

The developer shall describe the countermeasures implemented in order to prevent potential side channels of the TOE in the intended operational environment. The side channel analysis as part of the evaluator’s vulnerability analysis shall determine whether side channel exist and are exploitable i.e. these countermeasures are effective.

The developer and the evaluator should consult the SFR and the security objectives they enforce in order to determine whether an unintended information flow bypass the TSF or not. The implementation of a symmetric message authentication code calculation will keep the confidentiality of the key but it may or may be not required to protect the confidentiality of the processed user data. Therefore the decision about bypass of the TSF by leaking information about the processed user data depends on the security objective enforced by the SFR.