Joint Interpretation Library

Assurance Continuity – Practical cases for Smart Cards and similar devices

Version 1.0
November 2017

This page is intentionally left blank

Table of contents

1 Introduction

1.1. Background

1 This document seeks to provide some practical examples to identify the cases where evaluation work previously performed need not be repeated in all circumstances although a certified TOE or its environment have been changed. This document focuses on the AVA class only.

2 The purpose of Assurance Continuity is to enable developers to provide assured products to the IT consumer community in a timely and efficient manner. The awarding of a certificate signifies that all necessary evaluation work has been performed to convince the evaluation authority that the TOE meets all the defined assurance requirements as grounds for confidence that an IT product or system meets its security objectives.

1.2. Terminology

3 For clarity, the following terms are used in this document as defined in [5]:

a) the *certified TOE* refers to the version of the TOE that has been evaluated and for which a certificate has been issued.
b) the *changed TOE* refers to a version that differs in some respect from the certified TOE.
c) the *maintained TOE* refers to a changed TOE that has undergone the maintenance process and to which the certificate for the certified TOE also applies. This signifies that assurance gained in the certified TOE also applies to the maintained TOE.
d) the *Impact Analysis Report* (IAR) refers to a report which records the analysis of the impact of changes to the certified TOE. The IAR is generated by the developer who is requesting an addition to a maintenance addendum.
e) *maintenance* refers to the process of recognizing that a set of one or more changes made to a certified TOE (or to aspects of the development environment) have not adversely affected assurance in that TOE.
f) *re-evaluation* refers to the process of recognizing that changes made to a certified TOE (or to other assurance measures) require independent evaluator activities to be performed in order to establish a new assurance baseline. Re-evaluation seeks to reuse results from a previous evaluation.

2 Assurance continuity paradigm and practical examples

4 This chapter recalls the assurance continuity paradigm as defined in [5] and provides practical examples of changes that will be qualified as minor and major.

2.1. Assurance continuity paradigm

5 Assurance continuity seeks to exploit the fact that as changes are made to a unique TOE identifier (e.g. version increment), resulting from a changed certified TOE or its environment, evaluation work previously performed need not be repeated in all circumstances. The assurance continuity paradigm therefore defines the processes for maintenance and re-evaluation such that each seeks to recognise previous evaluation work.

6 *Maintenance* refers to the process undertaken by a developer in order to have a changed TOE, listed in the maintenance addendum for that TOE. It must be demonstrated that the changes to the TOE, the IT environment and/or the development environment do not adversely affect the assurance baseline.

7 *Re-evaluation* refers to the evaluation of a changed TOE, such that the developer could not (or chooses not to) demonstrate that changes to the certified TOE do not affect the assurance baseline.

8 It is important to note that the *maintenance* process is not intended to provide assurance in regard to the resistance of the TOE to new vulnerabilities or attack methods discovered since the date of the initial certificate. Such assurance can only be gained through *re-evaluation*. *Maintenance* only considers the effect of TOE changes on the assurance baseline; it does not consider an evolving threat environment.

9 Both the *maintenance* and *re-evaluation* processes have an equivalent starting point: when a change is made to the *certified TOE*. This change might be a patch designed to correct a discovered flaw, an enhancement to a feature, the addition of a new feature, a clarification in the guidance documentation, or any other change to the certified TOE. The decision whether a maintenance or re-evaluation process is appropriate, which is equivalent to the decision whether a minor or major change took place, depends on the documented changes and the developer rationale in the IAR. The decision should be a result of an alignment between certification body and developer. In doubt also the evaluation body could be involved.

10 A *minor change* is one whose impact is sufficiently minimal that it does not affect the assurance to the extent that the evaluator activities need be independently reapplied (although the developer is expected to have tested the changes as part of his standard regression testing) or a change to the development environment in which the change can be shown to have no follow-on effect on the other assurance measures that were in place at the time of the original evaluation. By contrast, a change deemed *major* has an impact that is substantial enough that it affects the assurance (except as noted above for the development environment) and would consequently require independent reapplication of the evaluator activities. Therefore, only minor changes are addressed under maintenance, which

is performed solely by the developer, while major changes are addressed under re-evaluation, which is performed by the evaluator.

11 It is impossible to predict all possible changes to all possible TOEs and, therefore, to identify the impact of all possible changes (and whether a given possible change is minor or major). Consequently, there is no fixed method for identifying whether the security impact of a change is major or minor. The following chapter identifies practical examples of changes.

2.2. Practical Examples

12 This chapter describes few practical cases for changes done at hardware or software level and evaluated during IC or ICC level respectively. Each example is written in such a way that a brief description of the change is given and then based on the nature of the changes whether this should be qualified as a minor or a major change. Furthermore, the penetration tests that are envisioned are described.

13 The examples are organized into three subchapters, the first dealing with guidance change and the following ones with hardware and software changes.

2.2.1 Guidance changes

14 A functional change in the guidance documentation will be considered as minor and therefore no penetration tests will be required whereas change to a mandatory security recommendation in the guidance will be considered as major and therefore a minimum set of appropriate tests may be required.

2.2.2 Hardware changes

15 This chapter describes few practical cases for changes done at hardware level or on the Dedicated Software and evaluated during the hardware evaluation.

16 The Impact Analysis is delivered by the IC developer including a differential description from the design sources to confirm which parts of the implementation have been modified and/or a source code and build outputs (e.g., assembly listings) differentials to confirm which parts of the implementation have been modified when the Dedicated Software or the cryptographic library are concerned. The type of description shall be in a way enabling examination of the differences on the lowest level of design, if appropriate down to transistor level.

2.2.2.1 Functional extension

17 The considered change in this example is due to a functional extension in one hardware block or addition of one communication interface and it induces limited difference within RTL code but with full re-synthesis and new place and route.

Impact Analysis Report (IAR):

Due to the full re-synthesis and new place and route, the chip is physically as a complete new chip and this is therefore considered as a major change. Although functionality/concepts/interfaces may be equal, physical behavior, signal run times, related analogue behavior, perturbation and LFI vulnerability are expected to be different with relevance on overall security level. In this case the hardware block change is regardless.

18 Full testing will be required in such a case.

2.2.2.2 Limited change on RNG hardware block

19 The considered change in this example is due to a limited change on the RNG hardware block (limited difference within RTL code but with partial re-synthesis).

Impact Analysis Report (IAR):

If the change is located:

a) On interfaces of the RNG only: it can be considered as a minor one.

b) In the analogue logic, respectively the entropy source: it is considered as a major one.

Partial re-synthesis is a matter of the area affected with regard to size and also other modules involved. By default it is relevant for being a major change.

20 Based on the above description, no penetration tests would be required in case a) and statistical testing based on a Quality metric will be required in case b) whereas additional tests such as Physical testing (FIB), fault injection should be considered when appropriate.

2.2.2.3 Change through metal fix

21 The considered change in this example is due to limited metal fixes following ESD issues on VDD regulator or metal fix enabling a feature at functional level.

Impact Analysis Report (IAR):

Different cases may occur:

a) If the module is for example regarding interfaces, memory logic or other modules not computing/managing sensitive data or signals, it could be non-relevant and considered as minor change.

b) If the metal fix is on voltage regulation it could be relevant regarding perturbation on external voltage and information leakage and considered as major change.

c) Metal fix enabling a feature on functional level, it could be relevant regarding perturbation on external voltage and information leakage and considered as major change.

22 Based on the above description, perturbation testing by spiking / glitching and side-channel, no penetration tests would be required in case a), verification testing (when appropriate) would be required in case b), whereas additional tests should be considered in case c) when appropriate.

2.2.2.4 Change in NVM size

23 The considered change in this example is due to a different NVM memory size.

Impact Analysis Report (IAR):

1. The Impact Analysis is delivered by the IC manufacturer including a differential description from the design sources to confirm which parts of the implementation have been modified.
2. NVM memory size change is achieved by:
   a) blocking and this is therefore considered as minor change,
   b) a new module of different size with localized and limited new place/route and re-synthesis (only linked to NVM size change) and this is therefore considered as minor change.

24 Based on above description, no penetration tests will be required for minor changes whereas depending on the area affected and amount of changes affecting surrounding modules of the NVM, a change of the side channel leakage or fault injection resistance can be expected and would require to be tested.

2.2.2.5 Wafer production change

25 The considered change in this example is due to transfer of design sources from one wafer production facility to another.

Impact Analysis Report (IAR): The Impact Analysis is delivered by the IC manufacturer. A change in wafer production is typically considered as a major change.

26 A minimum set of penetration tests would be required: side channel analysis (at least use of metrics to demonstrate similar leakage for hardware cryptographic-core and CPU and equivalent resistance) and fault injection to identify any difference on related countermeasures.

2.2.2.6 Technology node shrink change

27 The considered change in this example is due to a technology node shrink, the design sources are transferred to a new technology node from one wafer production facility to another.

Impact Analysis Report (IAR):

A technology shrink with such design sources transferred to a new technology node will in general be “limited”, as a big step in the technology node will require a new design, however this change is considered as major.

28 Full testing is required as the shrink always impacts the behavior of the chip regarding its leakage, physical entropy source and fault injection tolerance as the physical characteristics of the chip change.

2.2.3 Examples of Software changes

29 This chapter describes few practical cases for changes done at software level and evaluated during composite evaluation.

30 The Impact Analysis is delivered by the IC or ICC product developer including source code differential to confirm that only out of scope parts of the implementation have been modified, build outputs (e.g., assembly listings) and toolchain versioning information.

2.2.3.1 Change of non-security relevant functionality

31 The considered change in this example is performed on non-security relevant functionality or not related to security decision.

Impact Analysis Report (IAR):

Change of non-security relevant functionality can be: functionality of code used for initialization/personalization, return codes, flow/checks for different return codes, correction of functional bug using patch mechanisms, non-security relevant command, additional application compliant with the security guidance (e.g. non-Payment application on Payment cards, basic Java Card applet/Multos application, native application, GP Issuer Security Domain, transmission protocol, Telco functionality). These changes can be considered as minor if no side effect during the security impact analysis is identified.

32 Based on the above description, no penetration tests will be required for minor changes.

2.2.3.2 Change of security relevant functionality

33 The considered change in this example is done on security relevant functionality.

Impact Analysis Report (IAR):

Change of security relevant functionality can be cryptographic implementation, security measures, flow of security checks (e.g., PIN verification), handling of assets, low level memory access (copy/write NVM, etc..). These changes are therefore considered as major.

34 A minimum set of penetration tests may be required: side channel (use of metrics to demonstrate equivalent resistance), fault injection (at least verification testing) and any further tests needed such as software attacks.

2.2.3.3 Code relocation within the same memory

35 The considered change in this example occurs after relocation of part of the Embedded Software without functional change.

Impact Analysis Report (IAR):

Relocation of code without functional change can be applications loaded in a different order and impacting the logical/physical address (within the same memory) of the application under certification, adding/changing non-security

related code for e.g., personalization, change of buffer sizes. These changes can be considered as minor if no side effect during the security impact analysis is identified.

36 Based on the above description, no penetration tests will be required for minor changes.

2.2.3.4 Code relocation in a different memory

37 The considered change in this example occurs after relocation of Embedded Software in a different memory (e.g. from ROM to EEPROM) without any source code change.

Impact Analysis Report (IAR):

Loading an application or a cryptographic library under certification in a different memory is considered as major.

38 A minimum set of verification testing may be required, however experience gained from the lab on several similar changes (similar products) could mitigate the potential impact and thus avoid these penetration tests.

2.2.3.5 Configuration parameter(s) change

39 The considered change in this example is due to different configuration parameter(s) for non-security relevant functionality which was/were not included in the previous evaluation without any change of the code.

Impact Analysis Report (IAR):

Change of configuration parameter(s) for non-security relevant functionality without any change of the code can be Java Card package AID change, MIFARE/DESFIRE on/off or Transmission protocol. These changes can be considered as minor if no side effect during the security impact analysis is identified.

40 Based on the above description, no penetration tests will be required for minor changes.

2.2.3.6 Similar product on similar IC reference

41 The considered change in this example is due to the use of a new IC reference with almost the same source code.

Security Impact Analysis:

This is almost the same product as the one originally certified but on a new IC reference, say from the same IC family - same physical layout - with only limited code changes due to the IC change. These changes are usually considered as major.

42 A minimum set of verification testing may be required however experience gained from the lab on several similar changes (similar products) could mitigate the potential impact and thus avoid these penetration tests.

2.2.3.7 Change in Flash Bootloader code

43 The considered change in this example is due to issues in Flash Bootloader code.

Impact Analysis Report (IAR):

A functional change in the Bootloader is relevant as it could open access points for perturbation and also side channel attacks, if this is not covered by accompanying reasonable further hardware and software means.

a) If the Flash Bootloader is applied in secure environment only and permanently blocked prior reaching phase 7 (delivery to the end-user) the change can be considered as minor.

b) If the Flash Bootloader is protected against fault injection/SCA and source code review done by the ITSEF associated with developer functional verification demonstrate there is no security impact, the change would be considered as minor.

c) If there is no justified protection and the change implements e.g. cryptographic calculation, address-depending jumps etc. it is therefore considered as major change.

44 Based on above description, no penetration tests will be required for minor changes, cases a) and b). Else perturbation and side channel attacks should be considered for case c). For example, perturbation could block required security settings / configurations at start-up, software handling with secrets and address-depending jumps could be subject of SPA.

2.2.3.8 Change in cryptographic library code

45 The considered change in this example is due to functional issue in the cryptographic library code as for example an RSA key length update.

Impact Analysis Report (IAR):

A functional change in the cryptographic library code is relevant as it could open access points for failure and side channel analysis. RSA key length update by itself is not that relevant however this might have an impact on the efficiency of data randomization and/or blinding of exponents and it is therefore considered as major.

46 Based on above description, full testing might not always be necessary and verification testing could be considered sufficient.

3 References

[1] **Common Criteria.** *Common Methodology for Information Technology Security Evaluation v3.1 rev4.* September 2012

[2] **SOG-IS.** *SOGIS - IT technical domain v0.93.* February 2011

[3] **JHAS**. *Application of Attack Potential to Smartcards. v2.9.* January 2013

[4] **JHAS.** *Attack Methods for Smartcards and Similar Devices. v2.2.* January 2013

[5] **ASSURANCE CONTINUITY** - *CCRA REQUIREMENTS v2.1 June 2012*