**Common**

**Security Evaluation & Certification**

**Consortium**

**Common.SECC**

**Rule Book 1.6**

**Attachment 7 to Annex 3**

**Evaluation Technical Report (ETR) – Attack Scenarios and**

**Evaluator Tests ‑**

Version 1.1

29November, 2018

**Attack Scenarios and**

**Evaluator Tests**

as part of the

Evaluation Technical Report

**ETR-Part AVA:**

**Evaluation Assurance Level** **EAL POI**

**Version:**
##Version

**Date:**
##Date

**Filename:**
Template\_TOEName\_AttMethods\_yymmdd\_v10.docx

**Product:**
##TOE name (long)

**Developer:**
##sponsor (short)
##Address sponsor

**Evaluation Facility:**
##Evaluation facility

**Registration ID:**
##Registration ID

**Evaluator:**
name Evaluators

**Quality assurance:**

name QA

The following document is a template

Black text must be used without change. Especially headlines including the numbering of the headlines must not be changed.

Placeholders are marked in red colour and tagged with ##. The evaluator shall replace the placeholders with the actual value regarding the TOE consistently throughout all Single Evaluation Reports and documents.

The evaluator shall edit, if necessary, the red marked text and then change the colour to black. The green text must be considered by the evaluator and has to be deleted in the final version of the document.

Document Information

History of changes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Approved | Changes | Application Note (Reason for change, effects of change on work units, if applicable which comments of certification body were observed) |
| 1.00 | 20.11.2018 | - | Initial version | Template generated |
| 1.01 | 29 November 2018 |  | Endorsed by Common.SECC |  |

**Document invariants**

|  |  |  |
| --- | --- | --- |
| Name | Invariant (edit here) | Output value |
| Filename and size | calculated automatically | Template\_TOEName\_AttMethods\_yymmdd\_v10.docx |
| Current version | ##Version | ##Version |
| Date | ##Date | ##Date |
| Classification | Company Confidential | Company Confidential |
| TOE name (long) | ##TOE name (long) | ##TOE name (long) |
| TOE name (short) | ##TOE name (short) | ##TOE name (short) |
| Sponsor (long) | ##sponsor (long) | ##sponsor (long) |
| Sponsor (short) | ##sponsor (short) | ##sponsor (short) |
| Developer (short) | ##Developer (short) | ##Developer (short) |
| Evaluation facility | ##Evaluation facility | ##Evaluation facility |
| Registration ID | ##Registration ID | ##Registration ID |
| Certification body (long) | Common Security Evaluation and Certification Consortium | Common Security Evaluation and Certification Consortium |
| Certification body (short) | Common.SECC | Common.SECC |

**Content**

[1 Introduction 8](#_Toc530667066)

[2 Summary Table 8](#_Toc530667067)

[3 Attack Scenarios 11](#_Toc530667068)

[3.1 Minimal Invasive or Non-Invasive Attacks 11](#_Toc530667069)

[3.1.1 AS.1: Insert PIN Disclosing Bug on Flex PCB 11](#_Toc530667070)

[3.1.2 AS.2: Wire Hook Attacks 11](#_Toc530667071)

[3.1.3 AS.3: Attack Redundant Keys 11](#_Toc530667072)

[3.1.4 AS.4: Monitoring IC Card Supply 12](#_Toc530667073)

[3.1.5 AS.5: Monitoring Keyboard Scan Signal on Power Supply 12](#_Toc530667074)

[3.1.6 AS.6: Determine Keys by Side Channel Analysis 12](#_Toc530667075)

[3.2 Intrusion of Sensors, Switches and Filters 12](#_Toc530667076)

[3.2.1 AS.7: Removal Sensor Deactivation 12](#_Toc530667077)

[3.2.2 AS.9: Attack PCB Switch 12](#_Toc530667078)

[3.2.3 AS.10: Penetrate printed grids 12](#_Toc530667079)

[3.2.4 AS.11: Penetrate Potted Module 13](#_Toc530667080)

[3.2.5 AS.12: Penetration of the ICC reader 13](#_Toc530667081)

[3.3 Physical Attacks to Retrieve Secret Data 13](#_Toc530667082)

[3.3.1 AS.13: Probing into Secure Modules 13](#_Toc530667083)

[3.3.2 AS.14: Wire/finger/PCB routing Probing of Single Chip BGA 13](#_Toc530667084)

[3.3.3 AS.15: Chip Level Attacks 13](#_Toc530667085)

[3.3.4 AS.16: Physical Reading of Non Volatile Memories Content 13](#_Toc530667086)

[3.4 Perturbation Attacks 13](#_Toc530667087)

[3.4.1 AS.17: Perturbation Attacks on Lock Bits 13](#_Toc530667088)

[3.4.2 AS.18: Perturbation Attack on Software 13](#_Toc530667089)

[3.4.3 AS.19: DFA Caused by Transient Perturbations 13](#_Toc530667090)

[3.4.4 AS.20: Behaviour Analysis Attacks Caused by Transient Perturbations 13](#_Toc530667091)

[3.4.5 AS.21: Environmental Perturbation Attacks 14](#_Toc530667092)

[3.4.6 AS.22: Data Remanence of Volatile Memories Content 14](#_Toc530667093)

[3.5 Front Side Attacks 14](#_Toc530667094)

[3.5.1 AS.23: Cut Key Block 14](#_Toc530667095)

[3.5.2 AS.24: Rip Out Rubber Keys 14](#_Toc530667096)

[3.6 EMA and Sound Attacks 14](#_Toc530667097)

[3.6.1 AS.25: Monitoring Keyboard Sound 14](#_Toc530667098)

[3.6.2 AS.26: Monitoring Electromagnetic Emanation 14](#_Toc530667099)

[3.6.3 AS.27: Microwave Scanning 14](#_Toc530667100)

[3.6.4 AS.28: Monitoring Keyboard Emanation 14](#_Toc530667101)

[3.7 Attacks on RNG 14](#_Toc530667102)

[3.7.1 AS.29: Attack on True Random Number Generator (TRNG) 14](#_Toc530667103)

[3.7.2 AS.30: Attack on Pseudo Random Number Generator (PRNG) 15](#_Toc530667104)

[3.8 Software Attacks 15](#_Toc530667105)

[3.8.1 AS.31: Off-device Attacks 15](#_Toc530667106)

[3.8.2 AS.32: On-device Attacks 15](#_Toc530667107)

[3.9 PIN and Cryptographic Key Related Protocol Attacks 16](#_Toc530667108)

[3.9.1 AS.33: Limit Key Encryption Key search by Check Value 16](#_Toc530667109)

[3.9.2 AS.34: Weakly Padded PIN Blocks 16](#_Toc530667110)

[3.9.3 AS.35: Exhaustive PIN Search on Secondary PIN Related Functions 16](#_Toc530667111)

[3.9.4 AS.36: Clear Keypad Entry and Display Modification 16](#_Toc530667112)

[3.10 AS.37: Attack Magnetic Head Using Doors or Covers 16](#_Toc530667113)

[4 Penetration Tests 17](#_Toc530667114)

[4.1 General description 17](#_Toc530667115)

[4.2 ET.1: Gaining internal access 18](#_Toc530667116)

[4.2.1 Overview 18](#_Toc530667117)

[4.2.2 Removing adhesive Labels 18](#_Toc530667118)

[4.2.3 Opening detection switch 18](#_Toc530667119)

[4.2.4 Overcoming the case opening detection switch 18](#_Toc530667120)

[4.2.5 Bunker 18](#_Toc530667121)

[4.2.6 PSCR – Primary Smart Card Reader 18](#_Toc530667122)

[4.2.7 MSR 18](#_Toc530667123)

[4.2.8 Security Processor 19](#_Toc530667124)

[4.2.9 Keypad 19](#_Toc530667125)

[4.2.10 Display 19](#_Toc530667126)

[4.3 ET.2: Altering environmental conditions 20](#_Toc530667127)

[4.4 ET.3: Determining PIN digits by sound emission 21](#_Toc530667128)

[4.5 ET.4 Determining PIN digits EMA on key action 22](#_Toc530667129)

[4.6 ET.5: DPA and DEMA attack on AES and DES calculation 22](#_Toc530667130)

[4.6.1 DPA and SPA results 22](#_Toc530667131)

[4.6.2 DEMA and SEMA results 22](#_Toc530667132)

[4.7 ET.6: Installing a Bug in the ICC reader slot (widening for flex PCB or wire hook) 22](#_Toc530667133)

[4.8 ET.7: Analysis of Random Numbers 22](#_Toc530667134)

[4.9 ET.8: Design and code analysis 23](#_Toc530667135)

[4.9.1 Overview on architecture 23](#_Toc530667136)

[4.9.2 Security Commands of the Firmware of ##TOE name (long) 23](#_Toc530667137)

[4.9.3 Buffer Overflow 23](#_Toc530667138)

[4.9.4 Buffer Cleaning 23](#_Toc530667139)

[4.9.5 Firmware 23](#_Toc530667140)

[4.9.6 Conclusion 24](#_Toc530667141)

[4.10 Interface Penetration 24](#_Toc530667142)

[4.11 General code review 24](#_Toc530667143)

[4.11.1 Evaluator checks and verifications applicable for each product Add reference to current version of Common.SECC source code analysis document 24](#_Toc530667144)

[4.11.2 Evaluator checks and verifications typical for payment terminals 26](#_Toc530667145)

[4.11.3 Recommendations 28](#_Toc530667146)

[4.11.4 Evaluator checks and verifications applicable for each product 28](#_Toc530667147)

[4.11.5 Evaluator checks and verifications typical for payment terminals 30](#_Toc530667148)

[5 Annex 33](#_Toc530667149)

[5.1 Glossary and list of acronyms 33](#_Toc530667150)

[5.2 Bibliography 33](#_Toc530667151)

# Introduction

##sponsor (long) aims at completing a CC evaluation of the ##TOE name (short) terminal according the POI PP Add reference with the goal to have the ##TOE name (short) approved by the Common.SECC for the (UK and/or German) market. The evaluators have created this vulnerability assessment of the ##TOE name (short) and summarised the results in this document. According to [POI AttackPot] in this document the attack methods are applied to the ##TOE name (long) and an attack potential is determined if applicable. A summarizing list of all scenarios and ratings is given in chapter 2 whereas a detailed description of each scenario can be found in chapter 3. chapter 4 contains the actual penetration tests conducted by the evaluators.

A full security evaluation according to the Common Criteria methodology has been performed on parallel to this vulnerability assessment. Therefore, the results must be regarded as final. This report can be seen as Annex to the AVA Single Evaluation Report.

# Summary Table

| **Attack scenario** | **Corresponding Evaluator Tests** | **Relevant for TSF** | **Attack potential** |
| --- | --- | --- | --- |
| AS.1: Insert PIN Disclosing Bug on Flex PCB | ET.6 | PEDMiddleTSF | (not or) feasible/feasible within the required attack potential (16 points) |
| AS.2: Wire Hook Attacks | ET.6 | PEDMiddleTSF | not or feasible/feasible within the required attack potential (16 points) |
| AS.3: Attack Redundant Keys | ET.1 | - | … |
| AS.4: Monitoring IC Card Supply | ET.1 | - | not or feasible |
| AS.5: Monitoring Keyboard Scan Signal on Power Supply | - | CoreTSF | See AS.26 |
| AS.6: Determine Keys by Side Channel Analysis | ET.5 | CoreTSFKeys | … |
| AS.7: Removal Sensor Deactivation | ET.1 | - | … |
| AS.8: Case Switch Deactivation | ET.1 | CoreTSF | … |
| AS.9: Attack PCB Switch | - | CoreTSF | See AS.8 |
| AS.10: Penetrate printed grids | ET.1 | CoreTSF | not or feasible/feasible |
| AS.11: Penetrate Potted Module | - | CoreTSF | See AS.10 |
| AS.12: Penetration of the ICC reader | ET.6 | PEDMiddleTSF | not or feasible/feasible |
| AS.13: Probing into Secure Modules | - | CoreTSFKeysCoreTSFCoreTSF | Combination of AS.6 (not feasible within the required protection level (35 points)), AS.9 (not or feasible/feasible), AS.11 (not or feasible/feasible): not feasible |
| AS.14: Wire/finger/PCB routing Probing of Single Chip BGA | - | CoreTSFKeysCoreTSFCoreTSF | See AS.13 |
| AS.15: Chip Level Attacks | - | - | … |
| AS.16: Physical Reading of Non Volatile Memories Content | - | - | … |
| AS.17: Perturbation Attacks on Lock Bits | - | - | … |
| AS.18: Perturbation Attack on Software | - | - | … |
| AS.19: DFA Caused by Transient Perturbations | - | - | … |
| AS.20: Behaviour Analysis Attacks Caused by Transient Perturbations | - | - | … |
| AS.21: Environmental Perturbation Attacks | ET.2 | CoreTSFKeys | not or feasible/feasible |
| AS.22: Data Remanence of Volatile Memories Content | - | - | … |
| AS.23: Cut Key Block | - | CoreTSF | not or feasible/feasible |
| AS.24: Rip Out Rubber Keys | - | CoreTSF | not or feasible/feasible |
| AS.25: Monitoring Keyboard Sound | ET.3 | CoreTSF | … |
| AS.26: Monitoring Electromagnetic Emanation | ET.4 | CoreTSF | not or feasible/feasible |
| AS.27: Microwave Scanning | - | - | … |
| AS.28: Monitoring Keyboard Emanation | - | CoreTSF | See AS.26 |
| AS.29: Attack on True Random Number Generator (TRNG) | ET.7 | - | … |
| AS.30: Attack on Pseudo Random Number Generator (PRNG) | - | - | … |
| AS.31: Off-device Attacks | ET.8 | - | … |
| AS.32: On-device Attacks | ET.8 | - | … |
| AS.33: Limit Key Encryption Key search by Check Value | - | - | … |
| AS.34: Weakly Padded PIN Blocks | - | - | … |
| AS.35: Exhaustive PIN Search on Secondary PIN Related Functions | - | - | … |
| AS.36: Clear Keypad Entry and Display Modification | ET.1 | PEDMiddleTSF | not or feasible/feasible |
| AS.37: Attack Magnetic Head Using Doors or Covers | ET.1 | MSR | not or feasible/feasible |

# Attack Scenarios

## Minimal Invasive or Non-Invasive Attacks

### AS.1: Insert PIN Disclosing Bug on Flex PCB

The attack consists of the following steps:

1. Add step by step description of attack

This attack is (not or) feasible/feasible within the required attack potential (16 points) due to low protection level and the security measures applied in the device see sec. 4.2 and 4.7.

### AS.2: Wire Hook Attacks

In this attack scenario a wire is introduced via the card slot to contact the I/O pin of the ICC reader in order to collect plain-text PINs when they are sent to the card for offline verification. Since the card reader is enclosed by a tamper responsive boundary, a wire can only be introduced through the card slot. The card reader slot is placed at the front of the device. Therefore, hiding wires running out of the slot is considered to be difficult, due to it being in full view of the user in case of irregularities around the edges.

The attack consists of the following steps:

1. Add step by step description of attack

This attack is not or feasible/feasible within the required attack potential (16 points) due to low protection level and the security measures applied in the device see sec. 4.2 and 4.7.

### AS.3: Attack Redundant Keys

This attack is aimed against unused keys (or not frequently used, like backspace) in order to gain access to the key pad signal matrix and eavesdrop the PIN entry.

This attack consists of the following steps:

1. Add step by step description of attack

This attacks is not possible due to the construction of the keypad, refer to sec. 4.2.9.

### AS.4: Monitoring IC Card Supply

The power supply of the ICC Reader is equally protected as the I/O pin. An attack on the I/O pin is described in 4.2.6, PSCR – Primary Smart Card Reader

### AS.5: Monitoring Keyboard Scan Signal on Power Supply

Monitoring keyboard signals from outside of the device is not possible due to the various filters and voltage regulators that are in the path of the signal. Additionally, the components of the POI including the CPU produce a high level of noise which masks any keyboard activities. Reaching internal lines which might give access to cleaner signals is covered in ET.1: Gaining internal access.

An SPA attack on PIN digits is described in AS.26: Monitoring Electromagnetic Emanation.

### AS.6: Determine Keys by Side Channel Analysis

A side channel analysis attack on secret keys has been performed in ET.5: DPA and DEMA attack on AES and DES.

The attack aims at the determination of a DES key used for PIN encryption at the device using differential power analysis (DPA) and differential electromagnetic analyses (DEMA).

A function of the TOE is used to execute the DES calculation with the key under attack.

The attack consists of the following steps:

1. Add step by step description of attack

## Intrusion of Sensors, Switches and Filters

### AS.7: Removal Sensor Deactivation

The ##TOE name (long) is a Add description of performed attack

AS.8: Case Switch Deactivation

Add description of performed attack

### AS.9: Attack PCB Switch

Add description of performed attack

### AS.10: Penetrate printed grids

Add description of performed attack

### AS.11: Penetrate Potted Module

Add description of performed attack

### AS.12: Penetration of the ICC reader

Add description of performed attack

## Physical Attacks to Retrieve Secret Data

### AS.13: Probing into Secure Modules

Add description of performed attack

### AS.14: Wire/finger/PCB routing Probing of Single Chip BGA

Add description of performed attack

### AS.15: Chip Level Attacks

Add description of performed attack

### AS.16: Physical Reading of Non Volatile Memories Content

Add description of performed attack

## Perturbation Attacks

### AS.17: Perturbation Attacks on Lock Bits

Add description of performed attack

### AS.18: Perturbation Attack on Software

Add description of performed attack

### AS.19: DFA Caused by Transient Perturbations

Add description of performed attack

### AS.20: Behaviour Analysis Attacks Caused by Transient Perturbations

Add description of performed attack

### AS.21: Environmental Perturbation Attacks

**Voltage**

Add description of performed attack

**Temperature**

Add description of performed attack

### AS.22: Data Remanence of Volatile Memories Content

Add description of performed attack

## Front Side Attacks

### AS.23: Cut Key Block

Add description of performed attack

### AS.24: Rip Out Rubber Keys

Add description of performed attack

## EMA and Sound Attacks

### AS.25: Monitoring Keyboard Sound

Add description of performed attack

### AS.26: Monitoring Electromagnetic Emanation

Add description of performed attack

### AS.27: Microwave Scanning

Add description of performed attack

### AS.28: Monitoring Keyboard Emanation

Add description of performed attack

## Attacks on RNG

### AS.29: Attack on True Random Number Generator (TRNG)

Add description of performed attack

### AS.30: Attack on Pseudo Random Number Generator (PRNG)

Add description of performed attack

## Software Attacks

Add description of performed attack

### AS.31: Off-device Attacks

Add description of performed attack

#### Information Gathering

* Add description of performed analysis/attack

#### Editing Commands

* Add description of performed analysis/attack

#### Direct Protocols Attacks

* Add description of performed analysis/attack

#### Man-in-the-middle and Replay-Attacks

* Add description of performed analysis/attack

####  Buffer-Overflow Attacks

Add description of performed attack

### AS.32: On-device Attacks

Add description of performed attack

#### Secure Operating System

##### Direct Memory Accesses

Add description of performed attack

##### Illegal Access to buffers

Add description of performed attack

#### Hypervisor

Add description of performed attack

#### Virtual Machine

Add description of performed attack

## PIN and Cryptographic Key Related Protocol Attacks

### AS.33: Limit Key Encryption Key search by Check Value

Add description of performed attack

### AS.34: Weakly Padded PIN Blocks

Add description of performed attack

### AS.35: Exhaustive PIN Search on Secondary PIN Related Functions

Add description of performed attack

### AS.36: Clear Keypad Entry and Display Modification

Add description of performed attack

## AS.37: Attack Magnetic Head Using Doors or Covers

Add description of performed attack

# Penetration Tests

## General description

Add the general and technical description of the TOE here

## ET.1: Gaining internal access

The objective is to open the case in order to reach security relevant elements of the device, like …

### Overview

The back case is attached with 8 screws. It is not possible to open …

### Removing adhesive Labels

Labels are attached at back of the device. The label can be removed without …

### Opening detection switch

The device back is protected by …

### Overcoming the case opening detection switch

Removing the screws without prior preparative steps causes a tamper alert event immediately. An attacker has to overcome the security switch inside

### Bunker

The security Bunker is composed of …

**Conclusion:**

Due to the previously described construction of the bunker and secure mesh foil an attack is not (or is) feasible.

### PSCR – Primary Smart Card Reader

…

**Conclusion:**

Therefore an attack to gain access to the IO pin of the card reader is not or feasible.

### MSR

…

**Conclusion:**

Neither a second MSR head can be installed nor are the contacts of the head usable to gain information. Therefore this attack is not or feasible/feasible.

### Security Processor

…

Therefore attacks to gain access into the security processor are not or feasible/feasible.

### Keypad

…

**Conclusion:**

The combination of the applied countermeasures renders the attack on the keypad and its circuits as not or feasible/feasible.

### Display

…

**Conclusion:**

Therefore modifying display prompts that trick a user into entering his PIN when the POI software actually expects plain text entry is not or feasible/feasible

## ET.2: Altering environmental conditions

**Voltage Test and Temperature Test**

…

Conclusion:

The temperature protection system of the security processor is working reliably. A tamper event occurs before the device is out of its operational range and all secure data stored in the device is erased.

The attack is not or feasible/feasible.

## ET.3: Determining PIN digits by sound emission

…

**Mechanical sounds**

…

## ET.4 Determining PIN digits EMA on key action

…

**Conclusion**

…

This kind of attack is not or feasible/feasible.

## ET.5: DPA and DEMA attack on AES and DES calculation

### DPA and SPA results

…

### DEMA and SEMA results

…

**Conclusion**

…

## ET.6: Installing a Bug in the ICC reader slot (widening for flex PCB or wire hook)

…

This kind of attack is not or feasible/feasible.

## ET.7: Analysis of Random Numbers

…

**Conclusion**

…

## ET.8: Design and code analysis

### Overview on architecture

…

### Security Commands of the Firmware of ##TOE name (long)

…

### Buffer Overflow

To prevent Buffer Overflows there are mechanisms implemented in the Source Code. In the following the findings of a code analysis are described for the external and internal interfaces.

#### Smart Card Reader (ICCR)

Add logical analysis or result of source code review

* + - 1. **SAM Slots**

Add logical analysis or result of source code review

#### Magnetic Stripe Reader (MSR)

Add logical analysis or result of source code review

#### Keypad

Add logical analysis or result of source code review

### Buffer Cleaning

Add logical analysis or result of source code review

#### Secure PIN and Password entry

Add logical analysis or result of source code review

### Firmware

#### Self-Test

Add logical analysis or result of source code review

#### Secure FW Update

Add logical analysis or result of source code review

### Conclusion

No or An attack path has been found.

## Interface Penetration

The device exposes the following logic/physical interfaces:

* Wi-Fi / Ethernet
* Bluetooth
* USB / Serial interface
* …

All these interfaces were tested as follows:

Add result of logical interface penetration tests

## General code review

### Evaluator checks and verifications applicable for each product Add reference to current version of Common.SECC source code analysis document

|  |  |
| --- | --- |
| **No** | **General** |
| 1.
 | The evaluators ***shall check*** whether the implementation is well-structured and has sufficient comments in order that the evaluators can understand the implementation.Add analysis resultAdd the conclusion |
|  | The evaluators ***shall check*** compiler settings in order that test or debug settings are deactivated.Add analysis resultAdd the conclusion |
|  | The evaluators ***shall check*** whether third party software is used according to related security guidelines if such security guidelines are available.Add analysis resultAdd the conclusion |
| 1.
 | The evaluators ***shall verify*** the requirements of the underlying security hardware (setting and reading of security related flags, correct parameters of sensitive hardware calls, ...) according to the hardware security guideline. This holds especially for switches to turn on counter measures against side channel analysis or glitches.Add analysis resultAdd the conclusion |
|  | The evaluators ***shall verify*** the self-test mechanisms of the payment terminal by code analysis.Add analysis resultAdd the conclusion |
|  | The evaluators ***shall verify*** mechanisms to perform updates of the firmware by code analysis.Add analysis resultAdd the conclusion |

|  |  |
| --- | --- |
| **No** | The attacker discloses confidential data because of a **data buffer misuse**. |
|  | The evaluators ***shall check*** that dependent on the security architecture of the implementation any confidential data is not stored or used (as far as possible) in cleartext.Add analysis resultAdd the conclusion |
|  | The evaluators ***shall check*** that erasure of confidential data is done in a way that the compiler does not deactivate the erasure procedure. The corresponding risk is that the compiler may deactivate the erasure procedure because the deleted buffer is not used anymore afterwards. This holds for PINs (after verification) and private keys as well as symmetric keys (after usage).Add analysis resultAdd the conclusion |
|  | The evaluators ***shall check*** for external interfaces input/output buffer management (byte counters, boundary checks, ...). Alternatively penetration tests can be applied.Add analysis resultAdd the conclusion |

### Evaluator checks and verifications typical for payment terminals

|  |  |
| --- | --- |
| **No** | The attacker discloses confidential data or manipulates the security related process flow bya **timing attack**.Add analysis resultAdd the conclusion |
| 1.
 | The evaluators ***shall verify*** by code analysis that any verification of PINs and password is implemented timing independent. Thus e.g. the full data shall pass the verification procedure.Add analysis resultAdd the conclusion |
|  | The evaluators ***shall verify*** by code analysis that any verification of authentication codes (like MAC, …) is implemented timing independent. Thus e.g. the full data shall pass the verification procedure.Compiler optimization shall not lead to timing dependence (compiler flags to be checked).Add analysis resultAdd the conclusion |

|  |  |
| --- | --- |
| **No** | The attacker discloses confidential data because of a **data buffer misuse**. |
|  | The evaluators ***shall check*** by code analysis that any confidential data is explicitly erased after usage (e.g. by overwriting the buffer).This holds for PINs (after verification) and as well as private and symmetric keys (after usage).Add analysis resultAdd the conclusion |
|  | The evaluators ***shall check*** that well defined memory areas are used to store PINs, private keys and symmetric keys. Whenever possible pointers are to be used instead of copy them from one memory area to another.Add analysis resultAdd the conclusion |
|  | The evaluators ***shall check*** PIN block padding as well as PIN formatting by code analysis if random data is used for padding.Add analysis resultAdd the conclusion |

|  |  |
| --- | --- |
| **No** | The attacker discloses confidential data, manipulates sensitive data or manipulates the security related process flow **sending fuzzy data to the terminal**. |
|  | The evaluators ***shall identify*** at least functions processing cryptographic data, PINs, security verification results, security related counters, security related flags and PINs in the design specifications and ***shall check*** the implementation of these functions.Add analysis resultAdd the conclusion |

|  |  |
| --- | --- |
|  | The evaluators ***shall check*** that only specified commands are allowed to be implemented. This holds especially for smartcard APDU commands and online authorisation commands (if they are controlled by the platform).Add analysis resultAdd the conclusion |

|  |  |
| --- | --- |
| **No** | The attacker discloses confidential data, manipulates sensitive data or manipulates the security related process flow by **analysis of the random number generator (RNG)** |
|  | The evaluators ***shall check*** by code analysis whether the noise source (e.g. generated by a security processor) is processed in order that the evaluators are convinced that unpredictable random numbers are generated.Add analysis resultAdd the conclusion |
|  | The evaluators ***shall check*** that security functions use randoms only provided by RNGs examined by the code analysis.Add analysis resultAdd the conclusion |

### Recommendations

### Evaluator checks and verifications applicable for each product

|  |  |
| --- | --- |
| **No** | **General** |
|  | Compiler settings ***should be taken into consideration*** in order to see whether code optimization may lead to vulnerabilities in the executable code or the setting of warnings during compilation.Add analysis resultAdd the conclusion |
|  | The evaluators ***should check*** whether the used compiler creates vulnerabilities and whether e.g. a newer version not creating vulnerabilities exist.Add analysis resultAdd the conclusion |
|  | The evaluators ***should check*** that third party software is free of known vulnerabilities.Add analysis resultAdd the conclusion |
|  | Make files or similar configuration files ***should be taken into consideration*** in order to check whether only the intended source code files, libraries etc. are linked together to the final product.Add analysis resultAdd the conclusion |
|  | The evaluators ***should consider*** static code analysis tools or the vendors should use them and the evaluators ***should check*** the results of these tools.Add analysis resultAdd the conclusion |
|  | The evaluators ***should check*** the recommendations given in the CEM V3.1 R4, sections 1202 – 1204.Add analysis resultAdd the conclusion |

### Evaluator checks and verifications typical for payment terminals

|  |  |
| --- | --- |
| **No** | The attacker discloses confidential data, manipulates sensitive data or manipulates the security related process flow because of **internal illegal code instructions**.Add analysis resultAdd the conclusion |
|  | The evaluators ***should check*** security related case switches in order to verify that only the specified functions are called with the specified parameter. In particular a default block should be always present. This holds especially for the command processing and its parameter dependent called functions.Add analysis resultAdd the conclusion |
|  | The evaluators ***should consider*** drawing "trees" of main security functions with the called functions as branches in order to see which subroutines are called to implement the main security function. This holds especially for functions working on cryptographic keys and PINs. Add analysis resultAdd the conclusion |

|  |  |
| --- | --- |
|  | The evaluators ***should check*** whether the software is free of dead code or whether there is no branch to execute dead code.Add analysis resultAdd the conclusion |
|  | The evaluators ***should check*** whether return values reflect the behavior of the security related function.Add analysis resultAdd the conclusion |

|  |  |
| --- | --- |
| **No** | The attacker discloses confidential data, manipulates sensitive data or manipulates the security related process flow **sending fuzzy data to the terminal**. |
|  | The evaluators ***should check*** that processing of input data is well-structured. Add analysis resultAdd the conclusion |
|  | The evaluators ***should check*** based on functional specification that undocumented commands are not used. Add analysis resultAdd the conclusion |

|  |  |
| --- | --- |
| **No** | The attacker discloses confidential data, manipulates sensitive data or manipulates the security related process flow **performing a power interrupt**.Add analysis resultAdd the conclusion |
|  | The evaluators ***should check*** that sensitive data to be stored in non-volatile memory areas is written in a way being independent from a power interrupt.Add analysis resultAdd the conclusion |
|  | The evaluators ***should check*** that error counters aredecreased before operation and increased afterwards if no error appears.Add analysis resultAdd the conclusion |
|  | The evaluators ***should check*** that usage counters are increased before the operation, not afterwards.Add analysis resultAdd the conclusion |

# Annex

## Glossary and list of acronyms

|  |  |
| --- | --- |
| term | definition / explanation |
| … | … |
| … | … |

|  |  |  |
| --- | --- | --- |
| abbreviation | term | definition / explanation |
| ST | Security Target | … |
| … | … | … |
|  |  |  |

## Bibliography

Criteria and Methodology

[CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and general model, Sept 2012, Version 3.1, Revision 4, CCMB-2012-09-001

[CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security functional components, Sept 2012, Version 3.1, Revision 4, CCMB-2012-09-002

[CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security assurance components, Sept 2012, Version 3.1, Revision 4, CCMB-2012-09-003

[CEM] Common Methodology for Information Technology Security Evaluation, Evaluation methodology, Sept 2012, Version 3.1, Revision 4, CCMB-2012-09-004

Legislatives and Standards

Add reference or none

Evaluation Reports

Add reference

Developer Documents

Add reference

[ATECOV] Developer test coverage..

…

Other documents

[POI CEM] Joint Interpretation Library – CEM Refinements for POI Evaluation, Version 1.0, 27th May 2011. *Note: POI evaluations shall rely on the current version of this document at the moment of the evaluation.*

[POI AttackPot] Joint Interpretation Library / Application of Attack Potential to POIs, Version 1.92, 11th August 2014. *Note: POI evaluations shall rely on the current version of this document at the moment of the evaluation.*

[POI AttackMeth] Joint Interpretation Library / Attack Methods for POIs, Version 1.94, February 2015. *Note: POI evaluations shall rely on the current version of this document at the moment of the evaluation.*

Add reference