**Public**

# Infineon Technologies AG
## Chipcard & Security

Evaluation Documentation

# SLE88CNFX6600PM / m8864

# Security Target

**Version** **0.3**
**Date** **2011-05-30**
**Author** **Jürgen Noller**

| | |
|---|---|
| **Print Date:** | **6/17/2011 9:16:00 AM** |
| **Filename:** | **SLE88CNFX6600PM_SecTar.doc** |

# REVISION HISTORY

| | |
|---|---|
| 0.1 | 2010-12-20: Initial Version |
| 0.2 | 2011-02-25: Modification of FCS_COP.1 |
| 0.3 | 2011-05-30: Final Version |

# TABLE OF CONTENTS

## List of figures:



## List of tables

# 1 Security Target Introduction (ASE_INT)

## 1.1 Security Target and Target of Evaluation Reference

The title of this document is Security Target (ST) and comprises the Infineon Technologies Smart Card IC (Security Controller) SLE88CNFX6600PM, is internally registered under the development code m8864a13 and has the version number a13 with specific IC dedicated software.

The target of evaluation (TOE) SLE88CNFX6600PM / m8864a13 is described in the following. The Security Target has the revision 0.3 and is dated 2011-05-30.

The Target of Evaluation (TOE) is a smart card IC (Security Controller) as listed in Table 1 and its blocked derivatives listed in Table 3. The design step is a13.

The Security Target is based on the Protection Profile
"Smartcard IC Platform Protection Profile" [1].

The Protection Profile and the Security Target are built in compliance with Common Criteria v3.1.

The ST takes into account all relevant current final interpretations.

Table 1: Identification

| | Version | Date | Registration |
|---|---|---|---|
| Security Target | 0.3 | 2011-05-30 | M8864 A13 |
| Target of Evaluation<br>SLE88CNFX6600PM<br>SLE88CNFX6602PM<br>SLE88CNFX5400PM<br>SLE88CNF6600PM<br>SLE88CNF6602PM<br>SLE88CNF5400PM<br>SLE88CNFX6600P<br>SLE88CNFX6602P<br>SLE88CNFX5400P<br>SLE88CNF6600P<br>SLE88CNF6602P<br>SLE88CNF5400P<br>SLE88CFX6600P<br>SLE88CFX6602P<br>SLE88CFX5400P<br>SLE88CF6600P<br>SLE88CF6602P<br>SLE88CF5400P | a13 | | <br>m8864XXX a13<br>m8867XXX a13<br>m8960XXX a13<br>m8954XXX a13<br>m8957XXX a13<br>m8970XXX a13<br>m8865XXX a13<br>m8868XXX a13<br>m8961XXX a13<br>m8855XXX a13<br>m8958XXX a13<br>m8971XXX a13<br>m8866XXX a13<br>m8869XXX a13<br>m8962XXX a13<br>m8956XXX a13<br>m8959XXX a13<br>m8972XXX a13<br>all derivates with<br>PSL V3.22.11 and<br>guidance documentation |

| | | | |
|---|---|---|---|
| Guidance Documentation | Edition 2011-05<br>Edition 2009-12-19<br>Edition 2010-07-02 | 2011-05-31<br>2009-12-19<br>July 02,2010 | SLE88 Family - SLE88CNFXxxxxPM PSL & Security Reference Manual<br>SLE 88CNFX Family – Hardware Reference User´s Manual<br>SLE88CNFX Family Errata Sheet |
| Protection Profile | 1.0 | 2007-06-15 | Security IC Platform Protection Profile PP0035 |
| Common Criteria | Version 3.1 Revision 3 | 2009-July | Common Criteria for Information Technology Security Evaluation<br>Part 1: Introduction and general model CCMB-2009-07-001<br>Part 2: Security functional requirements CCMB-2009-07-002<br>Part 3: Security Assurance Components CCMB-2009-07-003 |

Remarks to the Target of Evaluation (TOE):

The TOE of this Security Target encloses the SLE88CNFX6600PM and seventeen different chip derivates. The hardware and the firmware of the SLE88CNFX6600PM and the seventeen derivates are identical (the version number is for all the a13 and the firmware version is for all the PSL V3.22.11, 660 kByte NVM, 256 kByte ROM and 32 kByte RAM). The differences between the derivates are the blocked crypto co-processor, the blocked contactless interfaces and the User NVM and User ROM size as shown in the following:

| | |
|---|---|
| SLE88CNFX6600PM: | 628 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CNFX6602PM: | 628 kByte User NVM, 24 kByte User RAM, 160 kByte User ROM |
| SLE88CNFX5400PM: | 508 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CNF6600PM: | 628 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CNF6602PM: | 628 kByte User NVM, 24 kByte User RAM, 160 kByte User ROM |
| SLE88CNF5400PM: | 508 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CNFX6600P: | 640 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CNFX6602P: | 640 kByte User NVM, 24 kByte User RAM, 160 kByte User ROM |
| SLE88CNFX5400P: | 520 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CNF6600P: | 640 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CNF6602P: | 640 kByte User NVM, 24 kByte User RAM, 160 kByte User ROM |
| SLE88CNF5400P: | 520 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CFX6600P: | 640 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CFX6602P: | 640 kByte User NVM, 24 kByte User RAM, 184 kByte User ROM |
| SLE88CFX5400P: | 520 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CF6600P: | 640 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |
| SLE88CF6602P: | 640 kByte User NVM, 24 kByte User RAM, 184 kByte User ROM |
| SLE88CF5400P: | 520 kByte User NVM, 24 kByte User RAM, 0 kByte User ROM |

The TOE, called SLE88CNFX6600PM in the following description, stands for the SLE88CNFX6602PM, SLE88CNFX5400PM, SLE88CNF6600PM, SLE88CNF6602PM, SLE88CNF5400PM, SLE88CNFX6600P, SLE88CNFX6602P, SLE88CNFX5400P, SLE88CNF6600P, SLE88CNF6602P, SLE88CNF5400P, SLE88CFX6600P, SLE88CFX6602P, SLE88CFX5400P, SLE88CF6600P, SLE88CF6602P and SLE88CF5400P.

The extension “M” in the product name denotes the Mifare™ [1] compatible Interface and the “N” denotes the near filed communication standard (NFC) interface. If the product name is neither including extension “M” nor “N”, it has the ISO7816 interfaces active only.

The derivates SLE88CFX6600P, SLE88CFX6602P, SLE88CFX5400P, SLE88CF6600P, SLE88CF6602P and SLE88CF5400P are the same hardware as the SLE88CNFX6600PM with the exception that the contactless communication via a near field communication standard (NFC) and the Mifare™ compatible Interface classic protocol is blocked during the production process. The blocking is done by deactivating during the production test without physically changing the TOE

The derivates SLE88CNF6600PM, SLE88CNF6602PM, SLE88CNF5400PM, SLE88CNF6600P, SLE88CNF6602P, SLE88CNF5400P, SLE88CF6600P, SLE88CF6602P and SLE88CF5400P are not providing the functionality of the Crypto@1408Bit coprocessor. Therefore the functionality of the Advanced Crypto Engine Driver and the patch loader mode of the Loader Filter Driver are not provided by these derivates as described in the [SoftwareManual]. The functionality RSA cryptography and the functionality patch loader mode (Loader Filter Driver) of the security enforcing function SF2 are not provided by these derivates. Not including this functionality has no impact of any other security policy of the TOE.

The blocking of the EEPROM is done by setting the according value in the chip configuration page, which is not available to the user. The same means of blocking are also used for switching on and off the accessibility of the cryptographic co-processor Crypto@1408Bit, the NFC and/or the Mifare™ compatible Interface protocol and also for the configuration of the ROM-sizes.

The memory settings are done during the production process by programming the physical start- and end-address of the user available memory areas. The entire configuration page including also the other blocking information can not be changed by the user afterwards and is protected against manipulation.

The firmware version PSL V3.22.11 can be tailored by the user to remove functionality, which the user decides not to use. The functionality of each tailored version is described in the guidance documentation of the TOE. The process to tailor a PSL version is described in the guidance documentation.

The TOE can be delivered to the user with PSL already tailored according to the user choice for the derivate SLE88CNFX6602PM, SLE88CNF6602PM, SLE88CNFX6602P, SLE88CNF6602P, SLE88CFX6602P and SLE88CF6602P.

For the derivates SLE88CNFX6600PM, SLE88CNFX5400PM, SLE88CNF6600PM, SLE88CNF5400PM, SLE88CNFX6600P, SLE88CNFX5400P, SLE88CNF6600P, SLE88CNF5400P, SLE88CFX6600P, SLE88CFX5400P, SLE88CF6600P and SLE88CF5400P, the TOE can be tailored by the user itself during his manufacturing process, as described in the guidance documentation.
A tailored PSL delivered on the TOE to the user does not include a code implementing functionality, which the user decided not to use. This, for example could be the SHA functionality, which is a part of SF6 according to P.Add-Functions. Not including the code implementing the SHA has no impact of any other security policy of the TOE, it is exactly equivalent to the situation where the user decides just not to use the SHA functionality.

A tailored PSL of the TOE could also for example exclude or deactivate the code implementing some security non enforcing functionality. Therefore this has no impact of any other security policy of the TOE.

[1] Mifare™ is a trademark of NXP B.V.

The PSL can be delivered completely stored on the TOE or a part of the PSL can be delivered in form of precompiled binary files (.obj). The filenames and the corresponding hash values are listed in section 11 Appendix.

## 1.2 Target of Evaluation overview

The TOE comprises the Infineon Technologies Smart Card IC (Security Controller) SLE88CNFX6600PM with specific IC dedicated software.

This Security Target (ST) describes the TOE known as the Infineon Technologies AG security controller group as listed in Table 1 and gives a summary product description.

The TOE is a member of the Security Controller family SLE88 and meets the highest requirements in terms of performance and security.
The TOE is intended to be used in smart cards for particularly high security-relevant applications. The TOE provides a real 32-bit CPU-architecture. The major components of the core system are the CPU (Central Processing Unit), the MMU (Memory Management Unit) and MED (Memory Encryption/Decryption Unit). The TOE implements a full linear addressable memory space for each privilege level, a simple scalable Memory Management concept.

The PSL software is providing additionally functionality via an API to the Smartcard Embedded Software. The STS firmware is used for test purposes during start-up and the Flash Loader allows downloading user software to the NVM during the manufacturing process or in the field.

The two cryptographic co-processors serve the need of modern cryptography: The DES module provides Triple-DES with dual-key or triple-key hardware acceleration. The Asymmetric Crypto Co-processor, called Crypto@1408Bit in the following, provides basic functions for RSA and Elliptic Curve (EC) cryptography.

In this security target the TOE is described and a summary specification is given. The security environment of the TOE during its different phases of the lifecycle is defined. The assets are identified which have to be protected through the security policy. The threats against these assets are described. The security objectives and the security policy are defined, as well as the security requirements. These security requirements are built up of the security functional requirements as part of the security policy and the security assurance requirements. These are the steps during the evaluation and certification showing that the TOE meets the targeted requirements. In addition, the functionality of the TOE matching the requirements is described.

The assets, threats, security objectives and the security functional requirements are defined in this Security Target and in [1] and are referenced here. These requirements build up a minimal standard common for all Smartcards.

The security functions are defined here in the security target as property of this specific TOE. Here it is shown how this specific TOE fulfils the requirements for the standard defined in the Protection Profile [1].

# 2 Target of Evaluation Description

The TOE description helps to understand the specific security environment and the security policy. In this context the assets, threats, security objectives and security functional requirements can be employed. The following is a more detailed description of the TOE than in [1] as it belongs to the specific TOE.

## 2.1 TOE Definition

The Target of Evaluation (TOE), the SLE88CNFX6600PM chip, is a smart card IC (Security Controller) meeting the highest requirements in terms of performance and security. It is manufactured by in a 0,13 µm CMOS technology. The IC is intended to be used in smart cards for particularly security-relevant applications. That is based on its previous use as developing platform for smart card operating systems according to the lifecycle model (in [1]).

The term “User Software” is used in the following for all operating systems and applications stored and executed on the TOE. The TOE is the platform for the user software. The user software itself is not part of the TOE.

The SLE88CNFX6600PM, whose block diagram is shown in Figure 2-1, consists of a dedicated microprocessor (CPU) with a virtual memory system (VMS), several different memories, security logic, a timer and an interrupt-controlled I/O interface, an interface management module, a Single Wire Protocol slave module (SWP) and two co-processors. The RNG module integrated on the chip consists of a TRNG (**T**rue **R**andom **N**umber **G**enerator).

The 32bit CPU is especially designed for smart card applications and provides powerful instructions for smart card applications. The memory comprises 32 KB of RAM, 184 KB of ROM and 660 KB of NVM. For the user 24 KB of RAM, 160 KB of ROM and 508 KB of NVM are minimal available depending on the derivate. It thus meets the requirements of the new generation of smartcard operating systems. The CPU accesses the memory via the integrated **M**emory **E**ncryption and **D**ecryption unit (MED). The access rights of the application to the memories can be controlled with the Virtual Memory System (VMS). Security, sleep mode and interrupt logic as well as the RNG are specially designed for smart card applications. The Sleep Mode logic (clock stop mode per ISO/IEC 7816-3) and the Supply-Shutdown Mode are used to reduce the overall power consumption. The timer permits easy implementation of communication protocols such as T=1 and all other time-critical operations. The input logic with uart-controlled I/O interface allows the smart card and terminal to be operated in parallel. The ICO unit of the input logic allows to operate the SLE88CNFX6600PM with a multiplication factor over the external clock signal or free running with maximum frequency. The Single Wire Protocol slave peripheral (SWP according to ETSI TS 102 613) connects an SWP master of the outside world via the SWP pad with the CPU of the TOE and contains a coprocessor for Mifare™ compatible Interface data stream de- and encryption. The TRNG provides random number data meeting high demands for e.g. cryptographic algorithms (keys) and protocols (challenges, blinding values, padding etc.).

Four modules for cryptographic operations are implemented on the TOE. The coprocessor Crypto@1408BIT is used for calculation of asymmetric algorithms like RSA or Elliptic Curve (EC) cryptography. This module is especially designed for chipcard applications with respect to the security and power consumption. The DES module computes the complete DES algorithm within a few clock cycles. That module is especially designed to counter attacks like DPA or EMA. The CRC module generates a 16-bit checksum conforming to ISO/IEC 3309 or CCITT V.41 to provide for the integrity of received or transmitted data and also calculates the parity bits on a byte granularity. Additionally the module SHA (Secure Hash Algorithm) is included as software module.

The firmware consists of two parts. The one is called platform support layer (PSL). It provides a convenient high level interface to the hardware devices like timers, UART (Universal Asynchronos Receiver Transmitter), Crypto@1408Bit, TRNG (Random Number Generator), NVM (Non Volatile

Memory), DES (Data Encryption Standard) and to the cryptographic functions AES (Advanced Encryption Standard), MD5, CRC (Cyclic Redundancy Check) and SHA (Secure Hash Algorithm).

The AES Encryption/Decryption Driver, the MD5 Generator Driver and the CRC Generator Driver are not in the scope of the certification and not part of the security features of the TOE.

The PSL provides the user (operating system) with the functionality to load code and data into the memory areas of the TOE in a secured process. The PSL hides all implementation specific details of a control operation performed at register level with a high security level of the implementation. The PSL is stored in ROM and NVM of the TOE. The use of the PSL is strongly recommended by the user guidance [SoftwareManual].

The other firmware part is the Self Test Software (STS), which controls the start-up of the chip. The STS configures all necessary parameters like keys for the MED. During the production test at the manufacturer the STS provides an interface to the test capabilities of the SLE88CNFX6600PM. The lock out of the test capabilities is also performed by the STS.

The TOE offers a new, improved standard of integrated security features, thereby meeting the requirements of all smart card applications such as information integrity, access control, mobile telephone, as well as uses in electronic funds transfer and healthcare systems.

To sum up, the TOE is a powerful smart card IC with a large amount of memory and special peripheral devices with both improved performance and optimized power consumption at minimal chip size. It therefore constitutes the basis for future smart card applications.

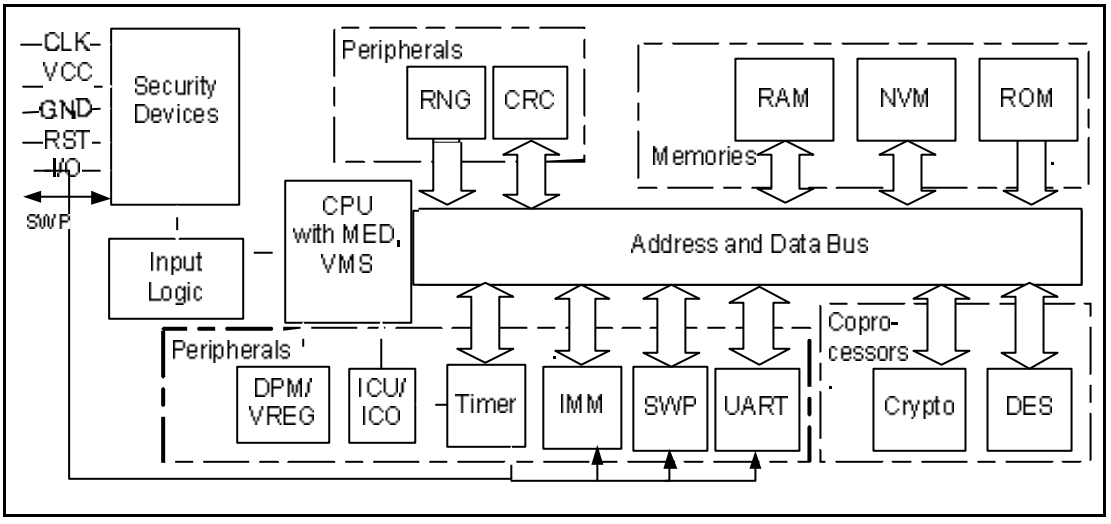



Figure 2-1: Block diagram of the SLE88CNFX6600PM hardware components

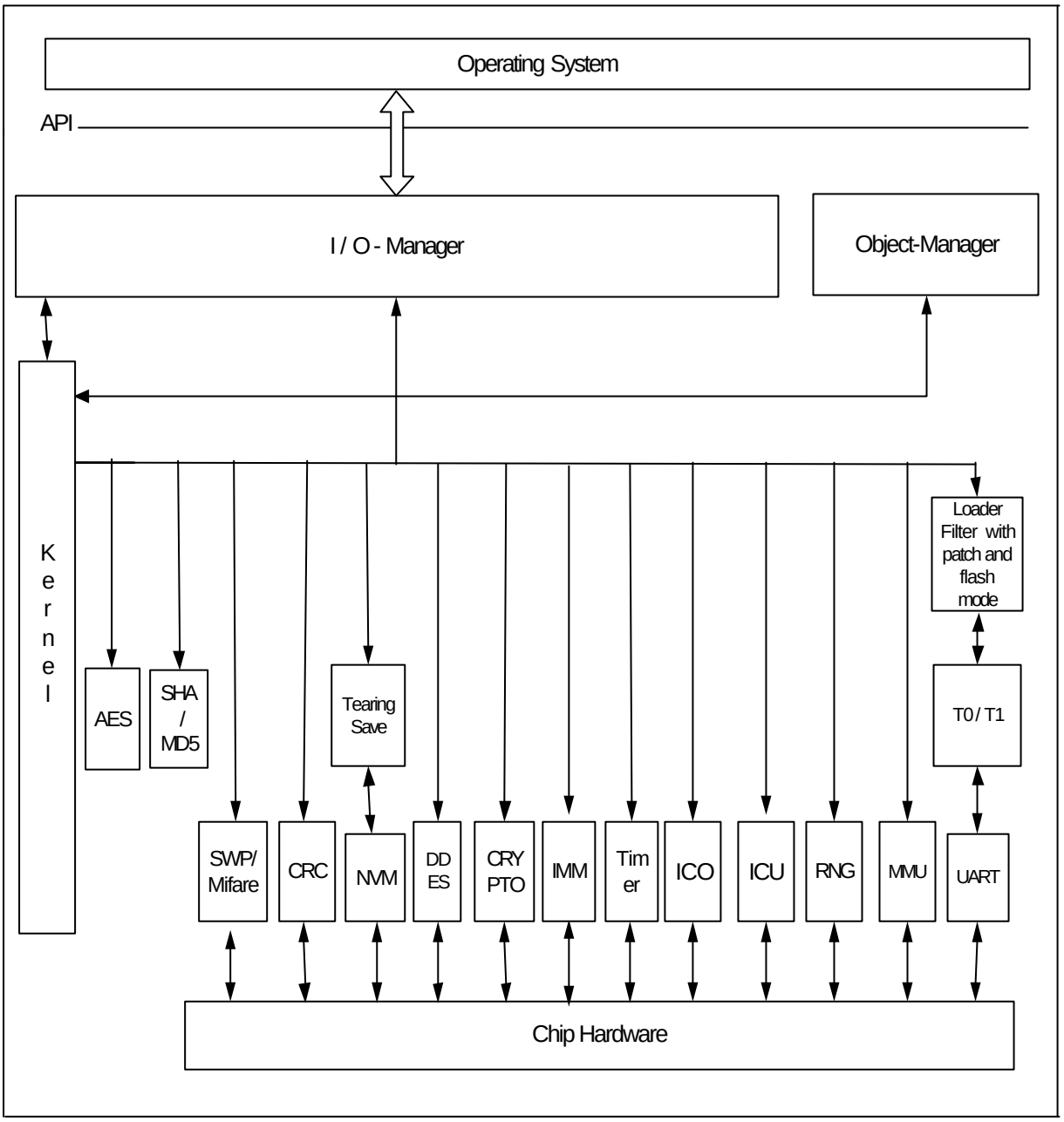



Figure 2-2: Block diagram of the SLE88CNFX6600PM Platform Support Layer (PSL)

## 2.2 Scope of the TOE

The TOE comprises the *hardware* of the smart card security controller, type SLE88CNFX6600PM, manufactured by Infineon, the associated *firmware* required for operation and provided in ROM and the associated software provided in ROM and NVM. The documents described in section 2.2.4 and listed in section 9.1 are supplied as a manual. In the following description, the term “manufacturer” is short term for the manufacturer of the TOE. The Smartcard Embedded Software respectively user software is not part of the TOE.

### 2.2.1 Hardware of the TOE

The *hardware part* of the TOE (cf. Figure 2-1) as defined in [1] is comprised of:

- Security devices comprising different sensors, filters, the selftest and the active shielding
- 32bit CPU with the subcomponents Memory Encryption and Decryption unit (MED3000) and Virtual Memory System (VMS) - includes countermeasures against side-channel attacks.
- Peripheral modules comprising:
  - Random Number Generator (RNG) including a true random number generator (TRNG) and a pseudo random number generator (PRNG)
    The PRNG is not in the scope of the certification and not part of the security features of the TOE
  - Interrupt module (ICU)
  - Timer (TIM)
  - Internal oscillator (ICO)
  - Voltage regulator (VREG)
  - Universal Asynchronous Receiver Transmitter (UART)
  - Dynamic Power management
  - Single Wire Protocol (SWP) slave peripheral including Mifare™ compatible Interface coprocessor
  - Interface Management Module (IMM)
  - CRC Module
- External memory comprising:
  - 32 KB extended RAM
  - 184 KB ROM, including the test routines (STS) and the PSL
  - 660 KB nonvolatile memory NVM.
- Cryptographic devices comprising:
  - Crypto@1408Bit for long integer modulo calculations, which are used in asymmetric algorithms like RSA
  - DES accelerator (DES), used for fast calculations of the DES algorithm - includes countermeasures against side-channel attacks
- Address and data bus (BUS)

Note 1:

The derivates SLE88CNF6600PM, SLE88CNF6602PM, SLE88CNF5400PM, SLE88CNF6600P, SLE88CNF6602P, SLE88CNF5400P, SLE88CF6600P, SLE88CF6602P and SLE88CF5400P are not providing the functionality of the Crypto@1408Bit cryptographic device.
End of note.

### 2.2.2 Firmware and software of the TOE

The entire software/firmware of the IC consists of two different parts. The one is the PSL as high level interface to the hardware functions (**P**latform **S**upport **L**ayer, IC Dedicated Support Software in [1]). The other part comprises the **S**elf **T**est **S**oftware (STS, IC Dedicated Support Software in [1] ). The STS consists of test and initialization routines and the Mifare™ compatible Interface data de-/encryption. The STS routines are not accessible for the user software due to VMS access rights.

The software part (PSL) of the TOE (cf. Figure 2-2) as defined in [1] is comprised of:

- IO-Manager
- Kernel
- Object-Manager
- MD5 Generator Driver
  The MD5 Generator Driver is not in the scope of the certification and not part of the security features of the TOE
- CRC Generator Driver (CRC)
- Secure Hash Algorithm Driver (SHA)
- AES Encryption/Decryption Driver (AES) - includes countermeasures against side-channel attacks.
  The AES Encryption/Decryption Driver is not in the scope of the certification and not part of the security features of the TOE
- NVM Driver (NVM)
- Tearing Save
- DDES Accelerator Driver (DDES)
- CRYPTO (Crypto@1408Bit) - includes countermeasures against side-channel attacks
- Timer Device Driver
- Memory Management Driver (MMU)
- Internal Clock Oscillator Driver (ICO)
- Interrupt Subsystem Driver (ICU)
- Random Number Generator Driver (RNG)
- UART
- T=0/T=1 Protocol Driver (T0/T1)
- Loader Filter Driver with patch loader and flash loader mode
- IMM Driver
- SWP Driver
- Mifare™ compatible Interface Driver

Note 2:

The derivates SLE88CNF6601PM, SLE88CNF6603PM, SLE88CNF5401PM, SLE88CNF6601P, SLE88CNF6603P, SLE88CNF5401P, SLE88CF6601P and SLE88CF6603P are not providing the functionality of the Crypto@1408Bit cryptographic device and the patch loader mode of the Loader

Filter Driver.
End of note.

The above demarcations of the TOE result in the interfaces described below.

### 2.2.3 Interfaces of the TOE

- The physical interface of the TOE to the external environment is the entire surface of the IC.
- The electrical interface of the TOE to the external environment is constituted by the pads of the chip, particularly the contacted RST, I/O0, CLK, SWP lines and supply lines VCC and GND.
- The data-oriented I/O interface to the TOE is formed by the I/O and SWP pads.
- The interface of the TOE to the operating system is constituted on the one hand by the PSL routine calls and on the other by the instruction set of the TOE.
- The interface of the TOE to the test routines is formed by the STS test routine call, i.e. entry to test mode (STS-TM entry).
- The derivates without extension “M” and “N” communicate with the contact-based interface according to ISO 7816/ETSI/EMV.
- The derivates with the extended “M” are used for the Mifare™ compatible Interface contactless interface protocol and related memory management (classic 1k emulation), but can communicate also contact based via ISO 7816.
- The derivates with the extended “N” are used for the Single-Wire-Protocol (SWP) protocol according [ETSI TS 102 613], but can communicate also contact based via ISO 7816.

### 2.2.4 Guidance documentation

The guidance documentation consists of the [HardwareManual], [SoftwareManual] and [ErrataSheet], which are containing the description of all interfaces of the software to the hardware relevant for programming the SLE88CNFX6600PM and the guidance to generate tailored PSL if necessary.

Finally the certification report will contain an overview of the recommendations to the software developer regarding the secure use of the platform SLE88CNFX6600PM. These recommendations are also included in the ordinary documentation.

The list of guidance documentation is given in section 9.1.

### 2.2.5 Forms of delivery

Several delivery processes exist during the lifecycle of the SLE88CNFX6600PM. The documentation and software development tools including the PSL are delivered from phase 2/3 to phase 1 in form of data carriers and paper documentation.

The SLE88CNFX6600PM can be delivered in form of complete modules, in form of plain wafers or in an IC case (e.g. DSO20). Additionally the SLE88CNFX6600PM (TOE) can be delivered finished or with an unfinished PSL software. In this case the delivery components are including an additionally part of the PSL software. The user has to implement this part of the PSL software during the personalization process of the operating system as described in the guidance documentation to finish the TOE. The delivery can therefore be at the end of phase 3 or at the end of phase 4 according to [1]. Nevertheless in all four cases the extended test features of the TOE are removed. In this document are always all four cases mentioned to avoid incorrectness but from the security policy point of view all four cases are identical.

### 2.2.6 Production sites

The TOE is produced in the production site Dresden. To distinguish different production sites the Chip Ident Mode data is coded as shown in Table 2.

The delivery measures are described in the ALC_DVS aspect.

Table 2: Production site in chip identification

| Production Site | Chip Identification (first nibble, hex format) |
|---|---|
| Dresden | 2 |

### 2.2.7 Availability of functionality

Table 3: Availability of functionality of the derivates:

| **TOE derivate** | **Crypto@ 1408Bit** | **ACE driver** | **Patch Loader** | **NFC** | **Mifare™** compatible Interface | **SF1 to SF7** |
|---|---|---|---|---|---|---|
| SLE88CNFX6600PM | activated | available | available | available | available | available |
| SLE88CNFX6602PM | activated | available | available | available | available | available |
| SLE88CNFX5400PM | activated | available | available | available | available | available |
| SLE88CNF6600PM | deactivated | not available | not available | available | available | Note 3 |
| SLE88CNF6602PM | deactivated | not available | not available | available | available | Note 3 |
| SLE88CNF5400PM | deactivated | not available | not available | available | available | Note 3 |
| SLE88CNFX6600P | activated | available | available | available | not available | available |
| SLE88CNFX6602P | activated | available | available | available | not available | available |
| SLE88CNFX5400P | activated | available | available | available | not available | available |
| SLE88CNF6600P | deactivated | not available | not available | available | not available | Note 3 |
| SLE88CNF6602P | deactivated | not available | not available | available | not available | Note 3 |
| SLE88CNF5400P | deactivated | not available | not available | available | not available | Note 3 |
| SLE88CFX6600P | activated | available | available | not available | not available | available |
| SLE88CFX6602P | activated | available | available | not available | not available | available |

| | | | | | | |
|---|---|---|---|---|---|---|
| SLE88CFX5400P | activated | available | available | not available | not available | available |
| SLE88CF6600P | deactivated | not available | not available | not available | not available | Note 3 |
| SLE88CF6602P | deactivated | not available | not available | not available | not available | Note 3 |
| SLE88CF5400P | deactivated | not available | not available | not available | not available | Note 3 |

Note 3:
The SF1, SF3 to SF5 and the SF7 are completely available for these derivates. For the SF2 the patch loader mode of the Loader Filter Driver is not available.
End of note.

The Crypto@1408Bit coprocessor is deactivated during the TOE life cycle phase 3. The production tests, which are done during the production (phase 3) as the TOE is in the test mode, are used to deactivate the Crypto@1408Bit coprocessor.

# 3 Conformance Claims (ASE_CCL)

## 3.1 CC Conformance Claim

This Security Target (ST) and the TOE claim conformance to Common Criteria version v3.1 part 1 [2], part 2 [3] and part 3 [4].

Conformance of this ST is claimed for:
Common Criteria part 2 extended and Common Criteria part 3 conformant.

## 3.2 PP Claim

This Security Target is in **strict conformance** to the
Security IC Platform Protection Profile [1].

The Security IC Platform Protection Profile is registered and certified by the Bundesamt für Sicherheit in der Informationstechnik[2] (BSI) under the reference BSI-PP-0035, Version 1.0, dated 15.06.2007.

The security assurance requirements of the TOE are according to the Security IC Platform Protection Profile [1]. They are all drawn from Part 3 of the Common Criteria version v3.1.

The augmentations of the PP [1] are listed below.

Table 4: Augmentations of the assurance level of the TOE

| Assurance Class | Assurance components | Description |
|---|---|---|
| Life-cycle support | **ALC_DVS.2** | Sufficiency of security measures |
| Vulnerability assessment | **AVA_VAN.5** | Advanced methodical vulnerability analysis |

## 3.3 Package Claim

This Security Target does not claim conformance to a package of the PP [1].

The assurance level for the TOE is EAL5 augmented with the components ALC_DVS.2 and AVA_VAN.5.

## 3.4 Conformance Rationale

This security target claims strict conformance only to one PP, the PP [1].

The Target of Evaluation (TOE) is a typical security IC as defined in PP chapter 1.2.2 comprising:

- the circuitry of the IC (hardware including the physical memories),
- configuration data, initialisation data related to the IC Dedicated Software and the behaviour of the security functionality
- the IC Dedicated Software with the parts
- the IC Dedicated Test Software,

[2] Bundesamt für Sicherheit in der Informationstechnik (BSI) is the German Federal Authority for Information Security

- the IC Dedicated Support Software.

The TOE is designed, produced and/or generated by the TOE Manufacturer.

Security Problem Definition:

Following the PP [1], the security problem definition is enhanced by adding an additional threat, an additional organizational security policy and an additional assumption. Including these add-ons, the security problem definition of this security target is consistent with the statement of the security problem definition in the PP [1], as the security target claimed strict conformance to the PP [1].

Conformance Rationale:

The augmented organizational security policy P.Add-Functions, coming from the additional security functionality of the cryptographic algorithms, the augmented assumption A.Key-Function, related to the usage of key-depending function, and the threat memory access violation T.Mem-Access, due to specific TOE memory access control functionality, have been added. These add-ons have no impact on the conformance statements regarding CC [2] and PP [1], with following rational:

- The security target remains conformant to CC [2], claim 482 as the possibility to introduce additional restrictions is given.
- The security target fulfils the strict conformance claim of the PP [1] due to the application notes 5, 6 and 7 which apply here. By those notes the addition of further security functions and security services are covered, even without deriving particular security functionality from a threat but from a policy.

Due to additional security functionality, one coming from the cryptographic functions - O.Add-Functions, and due to the memory access control - O.Mem-Access, additional security objectives has been introduced. These add-ons have no impact on the conformance statements regarding CC [2] and PP [1], with following rational:

- The security target remains conformant to CC [2], claim 482 as the possibility to introduce additional restrictions is given.
- The security target fulfils the strict conformance of the PP [1] due to the application note 9 applying here. This note allows the definition of high-level security goals due to further functions or services provided to the Security IC Embedded Software.

Therefore, the security objectives of this security target are consistent with the statement of the security objectives in the PP [1], as the security target claimed strict conformance to the PP [1].

The security requirements of this security target are consistent with the statement of the security requirements in the PP [1], as the security target claimed strict conformance to the PP [1].
All security functional requirements defined in the PP [1] are included and completely defined in this ST. The security functional requirements listed in the following are all taken from Common Criteria part 2 [3] and additionally included and completely defined in this ST:

- FDP_ACC.1 “Subset access control”
- FDP_ACF.1 “Security attribute based access control”
- FMT_MSA.1 “Management of security attributes”
- FMT_MSA.3 “Static attribute initialisation”
- FMT_SMF.1 “Specification of Management functions”
- FCS_COP.1 “Cryptographic support”
- FDP_SDI.2 “Stored data integrity monitoring and action”

The security functional requirement

- FPT_TST.2 “Subset TOE security testing“(Requirement from [3])

are included and completely defined in this ST, section 6.

All assignments and selections of the security functional requirements are done in the PP [1] and in this security target in section 7.2.

The Assurance Requirements of the TOE obtain the Evaluation Assurance Level 5 augmented with the assurance components ALC_DVS.2 and AVA_VAN.5 for the TOE.

# 4 Security Problem Definition (ASE_SPD)

The content of the PP [1] applies to this chapter completely.

## 4.1 Threats

The threats are directed against the assets and/or the security functions of the TOE. For example, certain attacks are only one step towards a disclosure of assets while others may directly lead to a compromise of the application security. The more detailed description of specific attacks is given later on in the process of evaluation and certification. An overview on attacks is given in PP [1] section 3.2.

The threats to security are defined and described in PP [1] section 3.2.

Table 5: Threats according PP [1]

| | |
|---|---|
| T.Phys-Manipulation | Physical Manipulation |
| T.Phys-Probing | Physical Probing |
| T.Malfunction | Malfunction due to Environmental Stress |
| T.Leak-Inherent | Inherent Information Leakage |
| T.Leak-Forced | Forced Information Leakage |
| T.Abuse-Func | Abuse of Functionality |
| T.RND | Deficiency of Random Numbers |

### 4.1.1 Additional Threat due to TOE specific Functionality

The additional functionality of introducing sophisticated privilege levels and access control allows the secure separation between the operation system(s) and applications, the secure downloading of applications after personalization and enables multitasking by separating memory areas and performing access controls between different applications. Due to this additional functionality “area based memory access control” a new threat is introduced.

The Smartcard Embedded Software is responsible for its User Data according to the assumption “Treatment of User Data (A.Resp-Appl)”. However, the Smartcard Embedded Software may comprise different parts, for instance an operating system and one or more applications. In this case, such parts may accidentally or deliberately access data (including code) of other parts, which may result in a security violation.

The TOE shall avert the threat “Memory Access Violation (T.Mem-Access)” as specified below.

T.Mem-Access Memory Access Violation

Parts of the Smartcard Embedded Software may cause security violations by accidentally or deliberately accessing restricted data (which may include code) or privilege levels. Any restrictions are defined by the security policy of the specific application context and must be implemented by the Smartcard Embedded Software.

Table 6: Additional threats due to TOE specific functions and augmentations

| | |
|---|---|
| T.Mem-Access | Memory Access Violation |

For details see PP [1] section 3.2.

### 4.1.2 Assets regarding the Threats

The primary assets concern the User Data which includes the user data as well as program code (Security IC Embedded Software) stored and in operation and the provided security services. These assets have to be protected while being executed and or processed and on the other hand, when the TOE is not in operation.

This leads to four primary assets with its related security concerns:

- SC1 Integrity of User Data and of the Security IC Embedded Software (while being executed/processed and while being stored in the TOE's memories),
- SC2 Confidentiality of User Data and of the Security IC Embedded Software (while being processed and while being stored in the TOE's memories)
- SC3 Correct operation of the security services provided by the TOE for the Security IC Embedded Software.
- SC4 Continuous availability of random numbers

SC4 is an additional security service provided by this TOE which is the availability of random numbers. These random numbers are generated either by a true random number or a deterministic random number generator or by both, when a true random number is used as seed for the deterministic random number generator. Note that the generation of random numbers is a requirement of the PP [1].

To be able to protect the listed assets the TOE shall protect its security functionality as well. Therefore critical information about the TOE shall be protected. Critical information includes:

- logical design data, physical design data, IC Dedicated Software, and configuration data
- Initialisation Data and Pre-personalisation Data, specific development aids, test and characterisation related data, material for software development support, and reticles.

The information and material produced and/or processed by the TOE Manufacturer in the TOE development and production environment (Phases 2 up to TOE Delivery) can be grouped as follows:

- logical design data,
- physical design data,
- IC Dedicated Software, Security IC Embedded Software, Initialisation Data and Pre-personalisation Data,
- specific development aids,
- test and characterisation related data,
- material for software development support, and
- reticles and products in any form

as long as they are generated, stored, or processed by the TOE Manufacturer.

For details see PP [1] section 3.1.

## 4.2 Organizational Security Policies

The TOE has to be protected during the first phases of their lifecycle (phases 2 up to TOE delivery which can be after phase 3 or phase 4). Later on each variant of the TOE has to protect itself. The organisational security policy covers this aspect.

P.Process-TOE Protection during TOE Development and Production

An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification.

The organisational security policies are defined and described in PP [1] section 3.3. Due to the augmentations of PP [1] an additional policy is introduced and described in the next chapter.

Table 7: Organizational Security Policies according PP [1]

| | |
|---|---|
| P.Process-TOE | Protection during TOE Development and Production |

### 4.2.1 Augmented Organizational Security Policy

Due to the augmentations of the PP [1] an additional policy is introduced.

The TOE provides specific security functionality, which can be used by the Smartcard Embedded Software. In the following specific security functionality is listed which is not derived from threats identified for the TOE's environment because it can only be decided in the context of the smartcard application, against which threats the Smartcard Embedded Software will use the specific security functionality.

The IC Developer / Manufacturer must apply the policy "Additional Specific Security Functionality (P.Add-Functions)" as specified below.

P.Add-Functions Additional Specific Security Functionality

The TOE shall provide the following specific security functionality to the Smartcard Embedded Software:

- Data Encryption Standard (DES),
- Triple Data Encryption Standard (3DES)
- Secure Hash Algorithm (SHA)

## 4.3 Assumptions

The TOE assumptions on the operational environment are defined and described in PP [1] section 3.4.

The assumptions concern the phases where the TOE has left the chip manufacturer.

A.Process-Sec-IC Protection during Packaging, Finishing and Personalization

It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).

A.Plat-Appl Usage of Hardware Platform

The Security IC Embedded Software is designed so that the requirements from the following documents are met: (i) TOE guidance documents (refer to the Common Criteria assurance class AGD) such as the hardware data sheet, and the hardware application notes, and (ii) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as documented in the certification report.

A.Resp-Appl Treatment of User Data

All User Data are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant User Data (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.

The support of cipher schemas needs to make an additional assumption.

Table 8: Assumption according PP [1]

| | |
|---|---|
| A.Process-Sec-IC | Protection during Packaging, Finishing and Personalization |
| A.Plat-Appl | Usage of Hardware Platform |
| A.Resp-Appl | Treatment of User Data |

### 4.3.1 Augmented Assumptions

The developer of the Smartcard Embedded Software must ensure the appropriate “Usage of Key-dependent Functions (A.Key-Function)” while developing this software in Phase 1 as specified below.

| | |
|---|---|
| A.Key-Function | Usage of Key-dependent Functions<br><br>Key-dependent functions (if any) shall be implemented in the Smartcard Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced). |

Note that here the routines which may compromise keys when being executed are part of the Smartcard Embedded Software. In contrast to this the threats T.Leak-Inherent and T.Leak-Forced address (i) the cryptographic routines which are part of the TOE

For details see PP [1] section 3.4.

# 5 Security objectives (ASE_OBJ)

This section shows the subjects and objects where are relevant to the TOE.
A short overview is given in the following.

The user has the following standard high-level security goals related to the assets:

- SG1 maintain the integrity of User Data and of the Security IC Embedded Software
- SG2 maintain the confidentiality of User Data and of the Security IC Embedded Software
- SG3 maintain the correct operation of the security services provided by the TOE for the Security IC Embedded Software
- SG4 provision of random numbers.

## 5.1 Security objectives for the TOE

The security objectives of the TOE are defined and described in PP [1] section 4.1.

Table 9: Objectives for the TOE according to PP [1]

| | |
|---|---|
| O.Phys-Manipulation | Protection against Physical Manipulation |
| O.Phys-Probing | Protection against Physical Probing |
| O.Malfunction | Protection against Malfunction |
| O.Leak-Inherent | Protection against Inherent Information Leakage |
| O.Leak-Forced | Protection against Forced Information Leakage |
| O.Abuse-Func | Protection against Abuse of Functionality |
| O.Identification | TOE Identification |
| O.RND | Random Numbers |

The TOE provides “Additional Specific Security Functionality (O.Add-Functions)” as specified below.

O.Add-Functions Additional Specific Security Functionality

The TOE must provide the following specific security functionality to the Smartcard Embedded Software:

- Data Encryption Standard (DES),
- Triple Data Encryption Standard (3DES),
- Secure Hash Algorithm (SHA).

The TOE shall provide “Area based Memory Access Control (O.Mem-Access)” as specified below.

O.Mem-Access Area based Memory Access Control

The TOE must provide the Smartcard Embedded Software with the capability to define restricted access memory areas. The TOE must then enforce the partitioning of such memory areas so that access of

software to memory areas and privilege levels is controlled as required, for example, in a multi-application environment.

Table 10: Additional objectives due to TOE specific functions and augmentations

| | |
|---|---|
| O.Add-Functions | Additional specific security functionality |
| O.Mem-Access | Area based Memory Access Control |

## 5.2 Security Objectives for the development and operational Environment

The security objectives for the security IC embedded software development environment and the operational environment is defined in PP [1] section 4.2 and 4.3. The table below lists the security objectives.

Table 11: Security objectives for the environment according to PP [1]

| | | |
|---|---|---|
| Phase 1 | OE.Plat-Appl | Usage of Hardware Platform |
| | OE.Resp-Appl | Treatment of User Data |
| Phase 5 – 6 optional Phase 4 | OE.Process-Sec-IC | Protection during composite product manufacturing |

### 5.2.1 Clarification of "Usage of Hardware Platform (OE.Plat-Appl)"

Regarding the cryptographic services this objective of the environment has to be clarified. The TOE supports cipher schemes as additional specific security functionality. If required the Smartcard Embedded Software shall use these cryptographic services of the TOE and their interface as specified. When key-dependent functions implemented in the Smartcard Embedded Software are just being executed, the Smartcard Embedded Software must provide protection against disclosure of confidential data (User Data) stored and/or processed in the TOE by using the methods described under "Inherent Information Leakage (T.Leak-Inherent)" and "Forced Information Leakage (T.Leak-Forced)".

The objectives of the environment regarding the memory, software and firmware protection and the SFR and peripheral-access-rights-handling have to be clarified. For the separation of different applications the Smartcard Embedded Software (Operating System) may implement a memory management scheme based upon security functions of the TOE.

### 5.2.2 Clarification of "Treatment of User Data (OE.Resp-Appl)"

Regarding the cryptographic services this objective of the environment has to be clarified. By definition cipher or plain text data and cryptographic keys are User Data. The Smartcard Embedded Software shall treat these data appropriately, use only proper secret keys (chosen from a large key space) as input for the cryptographic function of the TOE and use keys and functions appropriately in order to ensure the strength of cryptographic operation.

This means that keys are treated as confidential as soon as they are generated. The keys must be unique with a very high probability, as well as cryptographically strong. For example, it must be ensured that it is beyond practicality to derive the private key from a public key if asymmetric algorithms are used. If keys are imported into the TOE and/or derived from other keys, quality and confidentiality must be maintained. This implies that appropriate key management has to be realised in the environment.

Regarding the memory, software and firmware protection and the SFR and peripheral access rights handling these objectives of the environment has to be clarified. The treatment of User Data is also required when a multi-application operating system is implemented as part of the Smartcard Embedded Software on the TOE. In this case the multi-application operating system should not

disclose security relevant user data of one application to another application when it is processed or stored on the TOE.

### 5.2.3 Clarification of "Protection during Composite product manufacturing (OE.Process-Sec-IC)"

The protection during packaging, finishing and personalization includes also the personalization process (Flash Loader software) and the personalization data (TOE software components) during Phase 4, Phase 5 and Phase 6.

## 5.3 Security Objectives Rationale

The security objectives rationale of the TOE are defined and described in PP [1] section 4.4. For organizational security policy P.Add-Functions, OE.Plat-Appl and OE.Resp-Appl the rationale is given in the following description.

Table 12: Security Objective Rational

| Assumption, Threat or Organisational Security Policy | Security Objective |
|---|---|
| P.Add-Functions | O.Add-Functions |
| A.Key-Function | OE.Plat-Appl<br>OE.Resp-Appl |
| T.Mem-Access | O.Mem-Access |

The justification related to the security objective "Additional Specific Security Functionality (O.Add-Functions)" is as follows: Since O.Add-Functions requires the TOE to implement exactly the same specific security functionality as required by P.Add-Functions; the organisational security policy is covered by the objective.

Nevertheless the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced define how to implement the specific security functionality required by P.Add-Functions. (Note that these objectives support that the specific security functionality is provided in a secure way as expected from P.Add-Functions.) Especially O.Leak-Inherent and O.Leak-Forced refer to the protection of confidential data (User Data or TSF data) in general. User Data are also processed by the specific security functionality required by P.Add-Functions.

Compared to PP [1] clarification has been made for the security objective "Usage of Hardware Platform (OE.Plat-Appl)": If required the Smartcard Embedded Software shall use these cryptographic services of the TOE and their interface as specified. In addition, the Smartcard Embedded Software must implement functions which perform operations on keys (if any) in such a manner that they do not disclose information about confidential data. The non disclosure due to leakage A.Key-Function attacks is included in this objective OE.Plat-Appl. This addition ensures that the assumption A.Plat-Appl is still covered by the objective OE.Plat-Appl although additional functions are being supported according to O.Add-Functions.

Compared to the PP [1] a clarification has been made for the security objective "Treatment of User Data (OE.Resp-Appl)": By definition cipher or plain text data and cryptographic keys are User Data. So, the Smartcard Embedded Software will protect such data if required and use keys and functions appropriately in order to ensure the strength of cryptographic operation. Quality and confidentiality must be maintained for keys that are imported and/or derived from other keys. This implies that appropriate key management has to be realised in the environment. That is expressed by the assumption A.Key—Function which is covered from OE.Resp–Appl. These measures make sure that the assumption A.Resp-Appl is still covered by the security objective OE.Resp-Appl although additional functions are being supported according to P.Add-Functions.

Compared to the PP [1] an enhancement regarding memory area protection has been established. The clear definition of privilege levels for operated software establishes the clear separation of different restricted memory areas for running the firmware, downloading and/or running the operating system and to establish a clear separation between different applications. Nevertheless, it is also possible to define a shared memory section where separated applications may exchange defined data. The privilege levels clearly define by using a hierarchical model the access right from one level to the other. These measures ensure that the threat T.Mem-Access is clearly covered by the security objective O.Mem-Access.

The justification of the additional policy and the additional assumption show that they do not contradict to the rationale already given in the Protection Profile for the assumptions, policy and threats defined there.

# 6 Extended Component Definition (ASE_ECD)

There are four extended components defined and described for the TOE:

- the family **FCS_RNG** at the class FCS Cryptographic Support
- the family **FMT_LIM** at the class FMT Security Management
- the family **FAU_SAS** at the class FAU Security Audit
- the component **FPT_TST.2** at the class FPT Protection of the TSF

The extended components FCS_RNG, FMT_LIM and FAU_SAS are defined and described in PP [1] section 5. The component FPT_TST.2 is defined in the following.

## 6.1 Component "Subset TOE security testing (FPT_TST)"

The security is strongly dependent on the correct operation of the security functions. Therefore, the TOE shall support that particular security functions or mechanisms are tested in the operational phase (Phase 7). The tests can be initiated by the Smartcard Embedded Software and/or by the TOE or is done automatically and continuously.

Part 2 of the Common Criteria provides the security functional component "TSF testing (FPT_TST.1)". The component FPT_TST.1 provides the ability to test the TSF's correct operation.

For the user it is important to know which security functions or mechanisms can be tested. The functional component FPT_TST.1 does not mandate to explicitly specify the security functions being tested. In addition, FPT_TST.1 requires verification of the integrity of TSF data and of the stored TSF executable code which might violate the security policy. Therefore, the functional component **"Subset TOE security testing (FPT_TST.2)"** of the family TSF self test has been newly created. This component allows that particular parts of the security mechanisms and functions provided by the TOE are tested.

## 6.2 Definition of FPT_TST.2

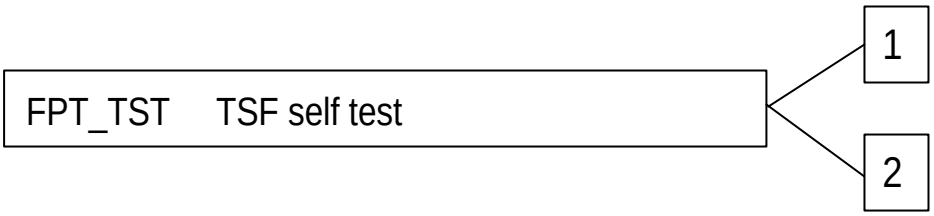
The functional component "Subset TOE security testing (FPT_TST.2)" has been newly created (Common Criteria Part 2 extended). This component allows that particular parts of the security mechanisms and functions provided by the TOE can be tested after TOE Delivery or are tested automatically and continuously during normal operation transparent for the user. This security functional component is used instead of the functional component FPT_TST.1 from Common Criteria Part 2. For the user it is important to know which security functions or mechanisms can be tested. The functional component FPT_TST.1 does not mandate to explicitly specify the security functions being tested. In addition, FPT_TST.1 requires verifying the integrity of TSF data and stored TSF executable code which might violate the security policy.

The functional component "Subset TOE testing (FPT_TST.2)" is specified as follows (Common Criteria Part 2 extended).

## 6.3 TSF self test (FPT_TST)

Family Behavior The Family Behavior is defined in [3] section 15.14 (442, 443).

Component levelling




FPT_TST.1: The component FPT_TST.1 is defined in [3] section 15.14 (444, 445, 446).

FPT_TST.2: Subset TOE security testing, provides the ability to test the correct operation of particular security functions or mechanisms. These tests may be performed at start-up, periodically, at the request of the authorized user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and executable code.

Management: FPT_TST.2

The following actions could be considered for the management functions in FMT:

- management of the conditions under which subset TSF self testing occurs, such as during initial start-up, regular interval or under specified conditions
- management of the time of the interval appropriate.

Audit: FPT_TST.2

There are no auditable events foreseen.

**FPT_TST.2** Subset TOE testing

Hierarchical to: No other components.

Dependencies: No dependencies

FPT_TST.2.1: The TSF shall run a suite of self tests [selection: during initial start-up, periodically during normal operation, at the request of the authorized user, and/or at the conditions [assignment: conditions under which self test should occur]] to demonstrate the correct operation of [assignment: functions and/or mechanisms].

# 7 Security Requirements (ASE_REQ)

For this section the PP [1] section 6 can be applied completely.

## 7.1 TOE Security Functional Requirements

The security functional requirements (SFR) for the TOE are defined and described in the PP [1] section 6.1 and in the following description.

The Table 13 provides an overview of the functional security requirements of the TOE, defined in the PP [1] section 6.1. In the last column it is marked if the requirement is refined. The refinements are also valid for this ST.

Table 13: Security functional requirements defined in PP [1]

| Security Functional Requirement | | Refined in PP [1] |
|---|---|---|
| FRU_FLT.2 | “Limited fault tolerance“ | Yes |
| FPT_FLS.1 | “Failure with preservation of secure state” | Yes |
| FMT_LIM.1 | “Limited capabilities” | No |
| FMT_LIM.2 | “Limited availability” | No |
| FAU_SAS.1 | “Audit storage” | No |
| FPT_PHP.3 | “Resistance to physical attack” | Yes |
| FDP_ITT.1 | “Basic internal transfer protection” | Yes |
| FPT_ITT.1 | “Basic internal TSF data transfer protection | Yes |
| FDP_IFC.1 | “Subset information flow control” | No |
| FCS_RNG.1 | “Quality metric for random numbers” | No |

The Table 14 provides an overview about the augmented security functional requirements, which are added additional to the TOE and defined in this ST. All requirements are taken from Common Criteria Part 2 [3], with the exception of the requirement FPT_TST.2, which is defined in this ST completely.

Table 14: Augmented security functional requirements

| Security Functional Requirement | |
|---|---|
| FPT_TST.2 | “Subset TOE security testing“ |
| FDP_ACC.1 | “Subset access control” |
| FDP_ACF.1 | “Security attribute based access control” |
| FMT_MSA.1 | “Management of security attributes” |
| FMT_MSA.3 | “Static attribute initialisation” |
| FMT_SMF.1 | “Specification of Management functions” |
| FCS_COP.1 | “Cryptographic support” |
| FDP_SDI.2 | “Stored data integrity monitoring and action” |

All assignments and selections of the security functional requirements of the TOE are done in PP [1] and in the following description.

The above marked extended components FMT_LIM.1 and FMT_LIM.2 are introduced in PP [1] to define the IT security functional requirements of the TOE as an additional family (FMT_LIM) of the Class FMT (Security Management). This family describes the functional requirements for the Test Features of the TOE. The new functional requirements were defined in the class FMT because this class addresses the management of functions of the TSF.

The additional component FAU.SAS is introduced to define the security functional requirements of the TOE of the Class FAU (Security Audit). This family describes the functional requirements for the storage of audit data and is described in the next chapter.

The requirement FPT_TST.2 is the subset of TOE testing and originated in [3]. This requirement is given as the correct operation of the security functions is essential. The TOE provides mechanisms to cover this requirement by the smartcard embedded software and/or by the TOE itself.

### 7.1.1 Extended Components FCS_RNG.1 and FAU_SAS.1

#### 7.1.1.1 FCS_RNG

To define the IT security functional requirements of the TOE an additional family (FCS_RNG) of the Class FCS (cryptographic support) is defined here. This family describes the functional requirements for random number generation used for cryptographic purposes.

| | |
|---|---|
| **FCS_RNG.1** | Random Number Generation |
| Hierarchical to: | No other components |
| Dependencies: | No dependencies |
| FCS_RNG.1 | Generation of random numbers requires that random numbers meet a defined quality metric. |
| FCS_RNG.1.1 | The TSF shall provide a physical random number generator that implements total failure test of the random source *and a continuous RNG test according to:*<br>*National Institute of Standards and Technology, Security Requirements for Cryptographic Modules, Federal Information Processing Standards Publication (FIPS) 140-2, 1999.* |
| FCS_RNG.1.2 | The TSF shall provide random numbers that meet *the functionality class P2 with SOF-high of [AIS31].* |

#### 7.1.1.2 FAU_SAS

To define the security functional requirements of the TOE an additional family (FAU_SAS) of the Class FAU (Security Audit) is defined here. This family describes the functional requirements for the storage of audit data. It has a more general approach than FAU_GEN, because it does not necessarily require the data to be generated by the TOE itself and because it does not give specific details of the content of the audit records.

The TOE shall meet the requirement “Audit storage (FAU_SAS.1)” as specified below (Common Criteria Part 2 extended).

| | |
|---|---|
| FAU_SAS.1 | Audit Storage |
| Hierarchical to: | No dependencies |
| Dependencies: | No dependencies. |
| FAU_SAS.1.1 | The TSF shall provide the test process *before TOE Delivery* with the capability to store *the Initialization Data and/or Pre-personalization* |

*Data and/or supplements of the Security IC Embedded Software* in the *not changeable configuration page area and non-volatile memory.*

### 7.1.2 Subset of TOE testing

The security is strongly dependent on the correct operation of the security functions. Therefore, the TOE shall support that particular security functions or mechanisms are tested in the operational phase (Phase 7). The tests can be initiated by the Smartcard Embedded Software and/or by the TOE.

The TOE shall meet the requirement "Subset TOE testing (FPT_TST.2)" as specified below (Common Criteria Part 2 extended).

| | |
|---|---|
| **FPT_TST.2** | Subset TOE testing |
| Hierarchical to: | No other components. |
| Dependencies: | No dependencies |
| FPT_TST.2.1 | The TSF shall run a suite of self tests *at the request of the authorised user* to demonstrate the correct operation of the *environmental mechanisms*[3]*.* |

### 7.1.3 Memory access control

Usage of multiple applications in one Smartcard often requires code and data separation in order to prevent that one application can access code and/or data of another application. For this reason the TOE provides Area based Memory Access Control. The underlying memory management unit (MMU) is documented in section 4 of the [HardwareManual].

The security service being provided is described in the Security Function Policy (SFP) **Memory Access Control Policy**. The security functional requirement **"Subset access control (FDP_ACC.1)"** requires that this policy is in place and defines the scope were it applies. The security functional requirement **"Security attribute based access control (FDP_ACF.1)"** defines security attribute usage and characteristics of policies. It describes the rules for the function that implements the Security Function Policy (SFP) as identified in FDP_ACC.1. The decision whether an access is permitted or not is taken based upon attributes allocated to the software. The Smartcard Embedded Software defines the attributes and memory areas. The corresponding permission control information is evaluated "on-the-fly" by the hardware so that access is granted/effective or denied/inoperable.

The security functional requirement **"Static attribute initialisation (FMT_MSA.3)"** ensures that the default values of security attributes are appropriately either permissive or restrictive in nature. Alternative values can be specified by any subject provided that the **Memory Access Control Policy** allows that. This is described by the security functional requirement **"Management of security attributes (FMT_MSA.1)"**. The attributes are determined during TOE manufacturing (FMT_MSA.3) or set at run-time (FMT_MSA.1).

From TOE's point of view the different roles in the Smartcard Embedded Software can be distinguished according to the memory based access control. However the definition of the roles belongs to the user software.

The following Security Function Policy (SFP) **Memory Access Control Policy** is defined for the requirement "Security attribute based access control (FDP_ACF.1)":

[3] The definition of the mechanisms can be found in the user guidance [SoftwareManual]

**Memory Access Control Policy**

The TOE shall control read and write accesses of software residing in memory areas on data including code stored in memory areas. The TOE shall restrict the ability to define, to change or at least to finally accept the applied rules (as mentioned in FDP_ACF.1) to software with the "privileged" attribute[4].

The memory model of the SLE88CNFX6600PM provides up to 255 different memory packages. The packages are divided in two classes. The one class consists of the privileged packages containing the operating system, the PSL and the SL. The other class is the class of regular packages containing different applications. The read and write access to packages may be allowed or denied by explicitly setting access rights. The access rights are split into accesses to the same package (intra-package) and between different packages (inter-package). The privileged packages do have complete access to regular packages.

The possible effective access rights (EAR) for data read/write operations are denoted in Table 15.

Table 15: Effective access rights (EAR) for data read/write operations

| Denotation | Intra access | Inter access |
|---|---|---|
| WW | Read / write | read / write |
| WR | Read / write | read / MPA |
| RR | Read / MPA | read / MPA |
| W- | Read / write | MPA / MPA |
| R- | Read / MPA | MPA / MPA |

Note that the MPA is the "Memory Protection Access Violation" trap.

The TOE shall meet the requirement "Subset access control (FDP_ACC.1)" as specified below.

**FDP_ACC.1** Subset access control

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACC.1.1 The TSF shall enforce the *Memory Access Control Policy* on *all subjects (software running at the defined and assigned privilege levels), all objects (data including code stored in memories) and all the operations defined in the Memory Access Control Policy*.

The TOE shall meet the requirement "Security attribute based access control (FDP_ACF.1)" as specified below.

**FDP_ACF.1** Security attribute based access control

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control
FMT_MSA.3 Static attribute initialisation

[4] The opposite of "privileged" is "regular"

| | |
|---|---|
| FDP_ACF.1.1 | The TSF shall enforce the *Memory Access Control Policy* to objects based on the following:<br>*Subject*:<br>- *software packages*<br>*Object*:<br>- *data including code stored in memories*<br>*Attributes*:<br>- *the logic memory area where the software is executed from and*<br>- *the logic memory area where the access is performed to and the operation (read or write) to be performed*. |
| FDP_ACF.1.2 | The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *evaluate the corresponding permission control information of the relevant memory range (EAR) before, during or after the access so that accesses to be denied can not be utilised by the subject (packages) attempting to perform the operation*. |
| FDP_ACF.1.3 | The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *none*. |
| FDP_ACF.1.4 | The TSF shall explicitly deny access of subjects to objects based on the *following additional rules: none*. |

The TOE shall meet the requirement “Static attribute initialisation (FMT_MSA.3)” as specified below.

| | |
|---|---|
| **FMT_MSA.3** | Static attribute initialisation |
| Hierarchical to: | No other components. |
| Dependencies: | FMT_MSA.1 Management of security attributes<br>FMT_SMR.1 Security roles |
| FMT_MSA.3.1 | The TSF shall enforce the *Memory Access Control Policy* to provide *well defined*[5] default values for security attributes that are used to enforce the SFP. |
| FMT_MSA.3.2 | The TSF shall allow the *“privileged” subjects* to specify alternative initial values to override the default values when an object or information is created. |

The TOE shall meet the requirement “Management of security attributes (FMT_MSA.1)” as specified below:

| | |
|---|---|
| **FMT_MSA.1** | Management of security attributes |
| Hierarchical to: | No other components. |
| Dependencies: | [FDP_ACC.1 Subset access control or<br>FDP_IFC.1 Subset information flow control]<br>FMT_SMF.1 Specification of management functions<br>FMT_SMR.1 Security roles |

[5] The static definition of the access rules is documented in [7]

FMT_MSA.1.1 The TSF shall enforce the *Memory Access Control Policy* to restrict the ability to *modify or delete* the security attributes *permission control information to the software with "privilege" attribute.*

The TOE shall meet the requirement "Specification of management functions (FMT_SMF.1)" as specified below:

**FMT_SMF.1** Specification of management functions

Hierarchical to: No other components

Dependencies: No dependencies

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions*: processing of the PSL function calls defined in [SoftwareManual], section 3.14 "Memory Management Unit Driver".*

### 7.1.4 Support of Cipher Schemes

The following additional specific security functionality is implemented in the TOE:

FCS_COP.1 Cryptographic operation requires a cryptographic operation to be performed in accordance with a specified algorithm and with a cryptographic key of specified sizes. The specified algorithm and cryptographic key sizes can be based on an assigned standard; dependencies are discussed in Section 7.3.1.1.

The following additional specific security functionality is implemented in the TOE:

- Data Encryption Standard (DES),
- Triple Data Encryption Standard (3DES)
- Secure Hash Algorithm (SHA)

**DES Operation**

The DES Operation of the TOE shall meet the requirement "Cryptographic operation (FCS_COP.1)" as specified below.

**FCS_COP.1/DES** Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/DES The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *Data Encryption Standard (DES)* in the *Electronic Codebook Mode (ECB)* and in the *Cipher Block Chaining Mode (CBC)* and with cryptographic key sizes *of 56 bit,* that meet the following: *standards*

*National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard (DES),*

*NIST Special Publication 800-67, Version 1.1*
*and*
*NIST Special Publication 800-38A, Edition 2001*

**Triple-DES Operation**

The DES Operation of the TOE shall meet the requirement "Cryptographic operation (FCS_COP.1)" as specified below.

**FCS_COP.1/3DES** Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/3DES The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *Triple Data Encryption Standard (3DES)* in the *Electronic Codebook Mode (ECB*) and in the *Cipher Block Chaining Mode (CBC)* and with cryptographic key sizes *of 2 x 56 bit or 3 x 56 bit,* that meet the following: *standards*

*National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard (DES), NIST Special Publication 800-67, Version 1.1*
*and*
*NIST Special Publication 800-38A, Edition 2001*

**SHA Operation**

The SHA Operation of the TOE shall meet the requirement "Cryptographic operation (FCS_COP.1)" as specified below.

**FCS_COP.1/SHA** Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1/SHA The TSF shall perform *hash value calculation of data* in accordance with a specified cryptographic algorithm *Secure Hash Standard (SHA)* and cryptographic key sizes of *a 160-bit and a 256-bit output* that meet the following: *standards*

*U.S. Department of Commerce, National Institute of Standards and Technology, Secure Hash Standard (SHA), FIPS PUB 180-3*

Note that the SHA cryptographic operation is a keyless operation.

Note 8
The secure hash-algorithm SHA is intended to be used for signature generation, verification and generic data integrity checks. The use for keyed hash operations like HMAC or similar, is not subject of this TOE and requires specific security improvements and DPA analysis by the operating system which is not part of this TOE.
End of note.

### 7.1.5 Data Integrity

The TOE shall meet the requirement “Stored data integrity monitoring and action (FDP_SDI.2)” as specified below:

| | |
|---|---|
| **FDP_SDI.2** | Stored data integrity monitoring and action |
| Hierarchical to: | FDP_SDI.1 stored data integrity monitoring |
| Dependencies: | No dependencies |
| FDP_SDI.2.1 | The TSF shall monitor user data stored in containers controlled by the TSF for *data integrity and one- and/or more-bit-errors* on all objects, based on the following attributes: *corresponding ECC value for RAM, ROM and EEPROM.* |
| FDP_SDI.2.2 | Upon detection of a data integrity error, the TSF shall *correct 1 bit errors in the RAM, ROM and EEPROM automatically and informs the user about not correctable errors on the request by the user.* |

## 7.2 TOE Security Assurance Requirements

The evaluation assurance level is EAL 5 augmented with ALC_DVS.2 and AVA_VAN.5. In the following table, the security assurance requirements are given. The augmentation of the assurance components compared to the Protection Profile [1] is expressed with bold letters.

Table 16: Assurance components

| Aspect | Acronym | Description | Refinement |
|---|---|---|---|
| **Development** | ADV_ARC.1 | Security Architecture Description | In PP [1] |
| | **ADV_FSP.5** | **Complete semi-formal functional specification with additional error information** | in ST |
| | ADV_IMP.1 | Implementation representation of the TSF | in PP [1] |
| | **ADV_INT.2** | **Well-structured internals** | |
| | **ADV_TDS.4** | **Semi-formal modular design** | |
| **Guidance Documents** | AGD_OPE.1 | Operational user guidance | in PP [1] |
| | AGD_PRE.1 | Preparative procedures | in PP [1] |
| **Life-Cycle Support** | ALC_CMC.4 | Production support, acceptance procedures and automation | in PP [1] |
| | **ALC_CMS.5** | **Development tools CM coverage** | in ST |
| | ALC_DEL.1 | Delivery procedures | in PP [1] |
| | ALC_DVS.2 | Identification of security measures | in PP [1] |
| | ALC_LCD.1 | Developer defined life-cycle model | |
| | **ALC_TAT.2** | **Compliance with implementation standards** | |
| **Security Target Evaluation** | ASE_CCL.1 | Conformance claims | |
| | ASE_ECD.1 | Extended components definition | |
| | ASE_INT.1 | ST introduction | |
| | ASE_OBJ.2 | Security objectives | |
| | ASE_REQ.2 | Derived security requirements | |
| | ASE_SPD.1 | Security problem definition | |
| | ASE_TSS.1 | TOE summary specification | |
| **Tests** | ATE_COV.2 | Analysis of coverage | in PP [1] |
| | **ATE_DPT.3** | **Testing: modular design** | |
| | ATE_FUN.1 | Functional testing | |
| | ATE_IND.2 | Independent testing - sample | |
| **Vulnerability Assessment** | AVA_VAN.5 | Advanced methodical vulnerability testing | in PP [1] |

### 7.2.1 Refinements

Some refinements are taken unchanged from the PP [1]. In some cases a clarification is necessary. In Table 16 an overview is given where the refinement is done.

Two refinements from the PP [1] have to be discussed here in the Security Target, as the assurance level is increased.

**Life cycle support (ALC_CMS)**

The refinement from the PP [1] can be applied even at the chosen assurance level EAL 5 augmented with ALC_CMS.5. The assurance package ALC_CMS.4 is extended to ALC_CMS.5 with aspects regarding the configuration control system for the TOE. The refinement is not touched.

**Functional Specification (ADV_FSP)**

The refinement from the PP [1] can be applied even at the chosen assurance level EAL 5 augmented with ADV_FSP.5. The assurance package ADV_FSP.4 is extended to ADV_FSP.5 with aspects regarding the descriptive level. The level is increased from informal to semi-formal with informal description. The refinement is not touched from this measure.

For details of the refinement see PP [1].

## 7.3 Security Requirements Rationale

### 7.3.1 Rationale for the Security Functional Requirements

The security functional requirements rationale of the TOE are defined and described in PP [1] section 6.3 for the following security functional requirements: FDP_ITT.1, FDP_IFC.1, FPT_ITT.1, FPT_PHP.3, FPT_FLS.1, FRU_FLT.2, FMT_LIM.1, FMT_LIM.2, FCS_RNG.1, and FAU_SAS.1.

The security functional requirements FPT_TST.2, FDP_ACC.1, FDP_ACF.1, FMT_MSA.1, FMT_MSA.3, FMT_SMF.1, FCS_COP.1 and FDP_SDI.2 are defined in the following description:

Table 17: Rational for additional SFR in the ST

| Objective | TOE Security Functional Requirements |
|---|---|
| O.Add-Functions | - FCS_COP.1/DES „Cryptographic operation“<br>- FCS_COP.1/3DES „Cryptographic operation”<br>- FCS_COP.1/SHA „Cryptographic operation“ |
| O.Phys-Manipulation | - FPT_TST.2 „ Subset TOE security testing “ |
| O.Mem-Access | - FDP_ACC.1 “Subset access control”<br>- FDP_ACF.1 “Security attribute based access control”<br>- FMT_MSA.3 “Static attribute initialisation”<br>- FMT_MSA.1 “Management of security attributes”<br>- FMT_SMF.1 “Specification of Management Functions” |
| O.Malfunction | - FDP_SDI.2 „Stored data integrity monitoring and action“ |

The table above gives an overview, how the security functional requirements are combined to meet the security objectives. The detailed justification is given in the following:

The justification related to the security objective “Additional Specific Security Functionality (O.Add-Functions)” is as follows:

The security functional requirement(s) “Cryptographic operation (FCS_COP.1)” exactly requires those functions to be implemented which are demanded by O.Add-Functions. Therefore, FCS_COP.1/DES, FCS_COP.1/3DES and FCS_COP.1/SHA are suitable to meet the security objective. The FCS_COP.1/SHA is a keyless algorithm and has no dependencies to FCS_CKM.1.

Nevertheless, the developer of the Smartcard Embedded Software must ensure that the additional functions are used as specified and that the User Data processed by these functions are protected as defined for the application context. These issues are addressed by the specific security functional requirements:

- [FDP_ITC.1 Import of user data without security attributes or
  FDP_ITC.2 Import of user data with security attributes or
  FCS_CKM.1 Cryptographic key generation],
- FCS_CKM.4 Cryptographic key destruction,

All these requirements have to be fulfilled by the environment to support OE.Resp-Appl for FCS_COP.1/DES and FCS_COP.1/3DES.

The security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced define how to implement the specific security functionality. However, key-dependent functions could be implemented in the Smartcard Embedded Software.

The usage of cryptographic algorithms requires the use of appropriate keys. Otherwise these cryptographic functions do not provide security. The keys have to be unique with a very high probability, and must have a certain cryptographic strength etc. In case of a key import into the TOE (which is usually after TOE delivery) it has to be ensured that quality and confidentiality are maintained. Keys for DES and 3DES have to be provided by the environment.

In this ST the objectives for the environment OE.Plat-Appl and OE.Resp-Appl have been clarified. The Smartcard Embedded Software defines the use of the cryptographic functions FCS_COP.1 provided by the TOE. The requirements for the environment FDP_ITC.1, FDP_ITC.2, FCS_CKM.1 and FCS_CKM.4 support an appropriate key management. These security requirements are suitable to meet OE.Resp-Appl.

The justification of the security objective and the additional requirements (both for the TOE and its environment) show that they do not contradict to the rationale already given in the Protection Profile for the assumptions, policy and threats defined there.

The security functional component Subset TOE security testing (FPT_TST.2) has been newly created (Common Criteria Part 2 extended). This component allows that particular parts of the security mechanisms and functions provided by the TOE can be tested after TOE Delivery. This security functional component is used instead of the functional component FPT_TST.2 from Common Criteria Part 2. For the user it is important to know which security functions or mechanisms can be tested. The functional component FPT_TST.2 does not mandate to explicitly specify the security functions being tested. In addition, FPT_TST.2 requires verification of the integrity of TSF data and stored TSF executable code which might violate the security policy.

The FPT_TST.2 tests parts of the security enforcing functions SF1 and SF3.

The security functional requirement FPT_TST.2 will detect attempts to conduce a physical manipulation on the monitoring functions of the TOE. The objective of FPT_TST.2 is O.Phys-Manipulation. The physical manipulation will be tried to overcome security enforcing functions.

The security functional requirement “Subset access control (FDP_ACC.1)” with the related Security Function Policy (SFP) “Memory Access Control Policy” exactly require the implementation of an area based memory access control as required by O.Mem-Access. The related TOE security functional requirements FDP_ACC.1, FDP_ACF.1, FMT_MSA.3, FMT_MSA.1 and FMT_SMF.1 cover this security objective. The implementation of these functional requirements is represented by the dedicated privilege level concept.

The justification of the security objective and the additional requirements show that they do not contradict to the rationale already given in the Protection Profile for the assumptions, policy and threats defined there. Moreover, these additional security functional requirements cover the requirements by [3] user data protection of chapter 11 which are not refined by the PP [1].

Nevertheless, the developer of the Smartcard Embedded Software must ensure that the additional functions are used as specified and that the User Data processed by these functions are protected as defined for the application context. The TOE only provides the tool to implement the policy defined in the context of the application.

The justification related to the security objective “Protection against Malfunction due to Environmental Stress (O.Malfunction)” is as follows:

The security functional requirement “Stored data integrity monitoring and action (FDP_SDI.2)” requires the implementation of an integrity observation and correction which is implemented by the Error Correction Control (ECC) measures. The ECC is present throughout all memories of the TOE while the ECC is realized in the EEPROM. These measures detect and inform about one and more bit errors at the request of the user. In case of 1 bit errors of the data are corrected automatically. By the ECC mechanisms it is prevented that the TOE uses corrupt data. Therefore FDP_SDI.2 is suitable to meet the security objective.

#### 7.3.1.1 Dependencies of Security Functional Requirements

The dependence of security functional requirements are defined and described in PP [1] section 6.3.2 for the following security functional requirements: FDP_ITT.1, FDP_IFC.1, FPT_ITT.1, FPT_PHP.3, FPT_FLS.1, FRU_FLT.2, FMT_LIM.1, FMT_LIM.2, FCS_RNG.1 and FAU_SAS.1.

The dependence of security functional requirements for the security functional requirements FPT_TST.2, FDP_ACC.1, FDP_ACF.1, FMT_MSA.1, FMT_MSA.3, FMT_SMF.1, FCS_COP.1 and FDP_SDI.2 are defined in the following description.

Table 18: Dependency for cryptographic operation requirement

| Security Functional Requirement | Dependencies | Fulfilled by security requirements |
|---|---|---|
| FCS_COP.1/DES | FCS_CKM.1 | Yes, see comment 2 |
| | FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1)<br>FCS_CKM.4 | Yes, see comment 2 |
| FCS_COP.1/3DES | FCS_CKM.1 | Yes, see comment 2 |
| | FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1)<br>FCS_CKM.4 | Yes, see comment 2 |
| FCS_COP.1/RSA | FCS_CKM.1 | Yes, see comment 2 |
| | FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1)<br>FCS_CKM.4 | Yes, see comment 2 |
| FCS_COP.1/SHA | FCS_CKM.1 and FCS_CKM.4 | Yes, see comment 3 |
| | FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1) | Yes, see comment 2 |
| FDP_ACC.1 | FDP_ACF.1 | Yes |
| FDP_ACF.1 | FDP_ACC.1<br>FMT_MSA.3 | Yes<br>Yes |
| FMT_MSA.3 | FMT_MSA.1<br>FMT_SMR.1 | Yes<br>Not required,<br>see comment 1 |
| FMT_MSA.1 | FDP_ACC.1 or FDP_IFC.1<br>FMT_SMR.1<br>FMT_SMF.1 | Yes<br>see comment 1<br>Yes |
| FMT_SMF.1 | None | N/A |
| FDP_SDI.2 | None | N/A |

Comment 1:
The dependency FMT_SMR.1 introduced by the two components FMT_MSA.1 and FMT_MSA.3 is considered to be satisfied because the access control specified for the intended TOE is not role-based but enforced for each subject. Therefore, there is no need to identify roles in form of a security functional requirement FMT_SMR.1.
End of comment.

Comment 2:
The security functional requirement “Cryptographic operation (FCS_COP.1)” met by the TOE has the following dependencies:

- [FDP_ITC.1 Import of user data without security attributes, or
- FDP_ITC.2 Import of user data with security attributes, or
- FCS_CKM.1 Cryptographic key generation]
- FCS_CKM.4 Cryptographic key destruction.

The security functional requirement “Cryptographic key management (FCS_CKM)” met by the TOE has the following dependencies:

- [FCS_CKM.2 Cryptographic key distribution, or
- FCS_COP.1 Cryptographic operation]

- FCS_CKM.4 Cryptographic key destruction.

These requirements all address the appropriate management of cryptographic keys used by the specified cryptographic function and are not part of the PP [1]. Most requirements concerning key management shall be fulfilled by the environment since the Smartcard Embedded Software is designed for a specific application context and uses the cryptographic functions provided by the TOE.
For the security functional requirement FCS_COP.1/DES and FCS_COP.1/3DES the respective dependencies FCS_CKM.1, FCS_CKM.4 and FDP_ITC.1 or FDP_ITC.2 have to be fulfilled by the environment. That mean, that the environment shall meet the requirements FCS_CKM.1 and FCS_CKM.4 as defined in [3], section 10.1 and shall meet the requirements FDP_ITC.1 or FDP_ITC.2 as defined in [3], section 11.7.
For the security functional requirement FCS_COP.1/SHA the respective dependencies FDP_ITC.1 or FDP_ITC.2 have to be fulfilled by the environment. That mean, that the environment shall meet the requirements FDP_ITC.1 or FDP_ITC.2 as defined in [3], section 11.7.

End of comment.

Comment 3:
The dependencies FCS_CKM.1 and FMT_CKM.4 are not required for the SHA algorithm, because the algorithm is a keyless operation. So the environment is not obligated to meet certain requirements for key management.
End of comment.

### 7.3.2 Rationale of the Assurance Requirements

The chosen assurance level EAL5 and the augmentation with the requirements ALC_DVS.2 and AVA_VAN.5 were chosen in order to meet the assurance expectations explained in the following paragraphs. In Table 18 the different assurance levels are shown as well as the augmentations. The augmentations are in compliance with the Protection Profile.

An assurance level EAL5 with the augmentations ALC_DVS.2 and AVA_VAN.5 are required for this type of TOE since it is intended to defend against **highly sophisticated attacks** without protective environment. This evaluation assurance package was selected to permit a developer to gain maximum assurance from positive security engineering based on good commercial practices. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defence against such attacks, the evaluators should have access to all information regarding the TOE including the TSF internals, the low level design and source code including the testing of the modular design. Additionally the mandatory technical document “Application of Attack Potential to Smartcards” [11] shall be taken as a basis for the vulnerability analysis of the TOE.

#### ALC_DVS.2 Sufficiency of security measures

Development security is concerned with physical, procedural, personnel and other technical measures that may be used in the development environment to protect the TOE.

In the particular case of a Security IC the TOE is developed and produced within a complex and distributed industrial process which must especially be protected. Details about the implementation, (e.g. from design, test and development tools as well as Initialization Data) may make such attacks easier. Therefore, in the case of a Security IC, maintaining the confidentiality of the design is very important.

This assurance component is a higher hierarchical component to EAL5 (which only requires ALC_DVS.1). ALC_DVS.2 has no dependencies.

**AVA_VAN.5 Advanced methodical vulnerability analysis**

Due to the intended use of the TOE, it must be shown to be highly resistant to penetration attacks. This assurance requirement is achieved by the AVA_VAN.5 component.

Independent vulnerability analysis is based on highly detailed technical information. The main intent of the evaluator analysis is to determine that the TOE is resistant to penetration attacks performed by an attacker possessing high attack potential.

AVA_VAN.5 has dependencies to ADV_ARC.1 “Security architecture description”, ADV_FSP.2 “Security enforcing functional specification”, ADV_TDS.3 “Basic modular design”, ADV_IMP.1 “Implementation representation of the TSF”, AGD_OPE.1 “Operational user guidance”, and AGD_PRE.1 “Preparative procedures”.

All these dependencies are satisfied by EAL5.

It has to be assumed that attackers with high attack potential try to attack Security ICs like smart cards used for digital signature applications or payment systems. Therefore, specifically AVA_VAN.5 was chosen in order to assure that even these attackers cannot successfully attack the TOE.

# 8 TOE Summary Specification (ASE_TSS)

The product overview is given in section 2.2. In the following the Security Features are described and the relation to the security functional requirements is shown.

The TOE is equipped with following Security Features to meet the security functional requirements:

SF1: Operating state checking

SF2: Phase management with test mode lock-out

SF3: Protection against snooping

SF4: TSF self test

SF5: Virtual Memory System (VMS)

SF6: Cryptographic support

SF7: NVM tearing save write

The following description of the Security Features is a complete representation of the TSF.

## 8.1 SF1: Operating state checking

Correct function of the SLE88CNFX6600PM is only given in the specified range of the environmental operating parameters. To prevent an attack exploiting that circumstance it is necessary to detect if the specified range is left.

All operating signals are filtered to prevent malfunction. The FRU_FLT.2 “Limited fault tolerance” requirement is satisfied.

In addition the operating state is monitored with sensors for the operating voltage, clock signal frequency, temperature and electro magnetic radiation (e.g. light). The TOE falls into the defined secure state in case of a specified range violation[6]. The defined secure state causes the chip internal reset process. The FPT_FLS.1 “Failure with preservation of secure state“ requirement is satisfied.

The TOE is equipped with an Error Correction Control mechanism (ECC) which covers the data in RAM, ROM and in non volatile memory (NVM). Thus introduced failures are securely detected by the ECC mechanism and, in terms of single bit errors in the non volatile memory also automatically corrected. In case of one and more bit errors the user can select one of several options (NMI, MI, Reset or don’t care) of reporting. The TOE therefore is protected by this mechanism against manipulation of memory content. The FDP_SDI.2 “Stored data integrity monitoring and action” is satisfied.

The covered security functional requirements are FRU_FLT.2 “Limited fault tolerance”, FPT_FLS.1 “Failure with preservation of secure state“ and FDP_SDI.2 “Stored data integrity monitoring and action”.

## 8.2 SF2: Phase management with test mode lock-out

The life cycle of the TOE is split-up in several phases. Chip development and production (phase 2, 3, 4) and final use (phase 4-7) is a rough split-up from TOE point of view. These phases are

[6] The operating state checking SF1 can only work when the TOE is running and can not prevent reverse engineering.

implemented in the SLE88CNFX6600PM as test mode (phase 2, 3, 4) and user mode (phase 1, 4-7). In addition a chip identification mode exists which is active in all phases.

During start-up of the SLE88CNFX6600PM the decision for the user mode or the test mode is taken dependent on several phase identifiers (phase management). If test mode is the active phase the SLE88CNFX6600PM requests authentication before any action (test mode lock-out). FMT_LIM.1 and FMT_LIM.2 are satisfied.

If the chip identification mode is requested the chip identification data (O.Identification) stored in a non modifiable EEPROM area is reported. FAU_SAS.1 “Audit storage” is satisfied.

During the production phase (phase 3) or after the delivery to the customer (phase 5 or phase 6), the TOE provides the possibility to load a user specific encryption key and user code and data encrypted into the empty (erased) NVM area as specified by the associated control information of the flash loader mode of the loader filter driver. The protocol of the loader filter driver ensures the confidentiality and integrity of the loaded code and data. After finishing the load operation, the flash loader mode is automatically deactivated, so that no second load operation with the flash loader mode is possible (FMT_LIM.2 “Limited availability”).
During the operation of the TOE the PSL provides the possibility to load signed code and data in the NVM and RAM areas as specified by the associated control information of the patch loader mode of the loader filter driver. The protocol of the loader filter driver ensures the confidentiality and integrity of the loaded code and data. The public part of the used signing key is stored in the NVM. This function could be deactivated permanently by the user software.

The code and data used by the Mifare™ compatible Interface protocol is located in the Security Layer (SL) of the TOE. The access to the Mifare™ compatible Interface protocol is restricted to the PSL to protect them against manipulation, disclosure and reconstruction. The FMT_LIM.1 is satisfied

The covered security functional requirements are FMT_LIM.1 “Limited capabilities”, FMT_LIM.2 “Limited availability” and FAU_SAS.1 “Audit storage”.

Note 9:

The derivates SLE88CNFX6601P, SLE88CNF6601P, SLE88CNFX6603P, SLE88CNF6603P, SLE88CNFX5401P, SLE88CNF5401P, SLE88CFX6601P, SLE88CF6601P, SLE88CFX6603P and SLE88CF6603P are not providing the functionality of the Mifare™ compatible Interface protocol. End of note.

## 8.3 SF3: Protection against snooping

Several mechanisms protect the SLE88CNFX6600PM against snooping the design or the user data during operation and even if it is out of operation (power down).

There are topological design measures for disguise, such as the use of the top metal layer “active shield” with active signals for protecting critical data. The entire design is kept in a non standard way to prevent attacks using standard analysis methods. A smartcard dedicated proprietary CPU with a non public bus protocol is used which makes analysis complicated.

The entire surface of the SLE88CNFX6600PM is protected with the active shield. Attacks over the surface are detected when the shield lines are cut or get contact.

The covered security functional requirement is FPT_PHP.3 “Resistance to physical attack“, as these measures make it difficult to do the physical analysis necessary before manipulation.

The readout of data can be controlled with the use of encryption. An attacker can not use the data he has espionage, because he must break the encryption.

The memory contents of the SLE88CNFX6600PM are encrypted on chip to protect against data analysis on stored data as well as on internally transmitted data. To prevent interpretation of leaked processed or transferred information additional randomness is inserted in the information.

In addition important parts of the CPU and the complete DES component are especially designed to counter leakage attacks like DPA or EMA. A special design method is used to make the current consumption nearly independent of the processed data.
The component is protected against information leakage.

The information leakage is kept low with special design measures. An interpretation of the leaked data is prevented as all the data is encrypted. The covered security functional requirements are FDP_ITT.1 “Basic internal transfer protection“ and FPT_ITT.1 “Basic internal TSF data transfer protection“. The encryption covers the data processing policy and FDP_IFC.1 “Subset information flow control“.

The covered security functional requirements are FPT_PHP.3 “Resistance to physical attack“, FDP_IFC.1 “Subset information flow control“, FDP_ITT.1 “Basic internal transfer protection“ and FPT_ITT.1 “Basic internal TSF data transfer protection“.

## 8.4 SF4: TSF self test

The TSF of the SLE88CNFX6600PM has either a hardware controlled self test which can be started from the user software or can be tested directly from the user software. The tested security enforcing functions are SF3 and only specific environmental mechanisms of SF1.

As any attempt to modify the sensor devices will be detected from the test, the covered security functional requirement is FPT_TST.2 “Subset TOE security testing“.

## 8.5 SF5: Virtual Memory System (VMS)

The VMS in the SLE88CNFX6600PM controls the address permissions of the privileged packages (memory areas) 1 and 2 and of the regular packages 3 to 15 and gives the software the possibility to define different access rights for the regular packages (memory areas) 16 to 255. The address permissions of the privileged package 0 are controlled by the hardware and the VMS. In case of an access violation the VMS will generate a trap. Then a trap service routine can react on the access violation. The policy of setting up the VMS and specifying the memory ranges for the regular packages 16 to 255 is defined from the user software in the upper layers. The two lower layers are given to the Security Layer (SL, layer 0) and to the Platform Support Layer (PSL, layer 1). The Operating system has the layer 2 and the Debug package has the layer 3. The layer 4 to 15 are not used and reserved for future use.

As the TOE provides support for separation of memory areas the covered security functional requirements are FDP_ACC.1 “Subset access control”, FDP_ACF.1 “Security attribute based access control”, FMT_MSA.3 “Static attribute initialization”, FMT_MSA.1 “Management of security attributes” and FMT_SMF.1 “Specification of Management functions”.

## 8.6 SF6: Cryptographic support

The TOE is equipped with several hardware accelerators and software modules to support the standard cryptographic operations. This security enforcing function is introduced to include the cryptographic operation in the scope of the evaluation as the cryptographic function itself is not used from the TOE security policy. On the other hand these functions are of special interest for the use of the hardware as platform for the software. The components are a combination of software and hardware unit to support DES and 3DES cryptography and software units to support the Secure Hash Algorithms (SHA).

**DES**

The TOE supports the encryption and decryption in accordance with the specified cryptographic algorithm Data Encryption Standard (DES) in the Electronic Codebook Mode (ECB) and in the Cipher Block Chaining Mode (CBC) and with cryptographic key sizes of 56 bit meeting the standard:

*National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard (DES), NIST Special Publication 800-67, Version 1.1* and *NIST Special Publication 800-38A, Edition 2001*

The covered security functional requirement is FCS_COP.1/DES.

**3DES**

The TOE supports the encryption and decryption in accordance with the specified cryptographic algorithm Triple Data Encryption Standard (3DES) in the Electronic Codebook Mode (ECB) and in the Cipher Block Chaining Mode (CBC) and with cryptographic key sizes of 112 bit or 168 bit meeting the standard:

*National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard (DES), NIST Special Publication 800-67, Version 1.1*and *NIST Special Publication 800-38A, Edition 2001*

The covered security functional requirement is FCS_COP.1/3DES.

**SHA**

The TOE supports hash-value calculation of chosen data in accordance with a specified cryptographic algorithm SHA and with cryptographic key sizes of none that meet the following standards:

*U.S. Department of Commerce, National Institute of Standards and Technology, Secure Hash Standard (SHA), FIPS PUB 180-3*

Regarding the SHA algorithm it has to be noted that the secure hash-algorithm SHA is intended to be used for signature generation, verification and generic data integrity checks. The use for keyed hash operations like HMAC or similar security critical operations involving keys, is not subject of this TOE and requires specific security improvements and DPA analysis including the operating system, which is not part of this TOE.

The covered security functional requirement is FCS_COP.1/SHA.

Random data is essential for cryptography as well as for physical security mechanisms. The SLE88CNFX6600PM is equipped with a true random generator (TRNG) based on physical probabilistic controlled effects. The random data can be used from the user software as well as from the security enforcing functions. The required tests defined in [AIS31] are provided from the PSL.

The generated numbers are of true random nature due to the construction principle of the RNG and fulfill the requirements of the class “P2 high” criteria specified in [AIS31].

The covered security functional requirement is FCS_RNG.1.

As defined the cryptographic operations are provided by the TOE, the covered security functional requirements are FCS_COP.1/DES, FCS_COP.1/3DES, FCS_COP.1/SHA and FCS_RNG.1.

## 8.7 SF7: NVM tearing safe write

The hardware of the NVM together with the PSL supports the TOE with a function to copy one data block with a defined maximum number of bytes or/and one or a bunch with a maximum number of data blocks of any data size to different NVM locations, under the protection of a data security mechanism. The data security mechanism keeps a backup copy of either the old or the new contents of all addressed NVM pages before they are overwritten. If the update of the data fails due to an unexpected card tearing, the old or the new contents of all target areas affected by the transaction is recovered at the next power-up.
As defined NVM tearing safe write operations are provided by the TOE, the covered requirements

are FRU_FLT.2 “Limited fault tolerance” and FPT_FLS.1 “Failure with preservation of secure state”. The NVM tearing safe write detects errors that happens during the NVM write operation and correct the errors to provide the correct function of the TOE. If a correction is not possible the TOE is forced into a secure state.

## 8.8 Assignment of Security Functional Requirements to TOE’s Security Functionality

The justification and overview of the mapping between security functional requirements (SFR) and the TOE’s security functionality (SF) is given in sections the sections above. The results are shown in Table 19. The security functional requirements are addressed by at least one relating security feature.

The security functional requirements FPT_FLS.1 and FRU_FLT.2 are covered mutually supportive from hardware SFs and software SFs. FPT_FLS.1 “Failure with preservation of secure state” and FRU_FLT.2 “Limited fault tolerance” are covered from the SF1 regarding the hardware aspects by filtering the external signals or resetting the TOE and SF7 regarding the software aspects by detecting erroneous states in NVM programming and to react with recovering a defined state.

The security functional requirement FPT_PHP.3 is covered from the SF3 for the aspect of making the reverse engineering harder even if the TOE is out of operation and for the aspect of detecting the attempt to modify the TOE when the chip is running. The SF3 is mutually supportive to cover FPT_PHP.3.

Table 19: Mapping of SFR and SF

| **Security Functional Requirement** | SF1 | SF2 | SF3 | SF4 | SF5 | SF6 | SF7 |
|---|---|---|---|---|---|---|---|
| FAU_SAS.1 | | X | | | | | |
| FCS_RNG.1 | | | | | | X | |
| FDP_IFC.1 | | | X | | | | |
| FDP_ITT.1 | | | X | | | | |
| FMT_LIM.1 | | X | | | | | |
| FMT_LIM.2 | | X | | | | | |
| FPT_FLS.1 | X | | | | | | X |
| FPT_ITT.1 | | | X | | | | |
| FPT_PHP.3 | | | X | | | | |
| FRU_FLT.2 | X | | | | | | X |
| FPT_TST.2 | | | | X | | | |
| FDP_ACC.1 | | | | | X | | |
| FDP_ACF.1 | | | | | X | | |
| FMT_MSA.3 | | | | | X | | |
| FMT_MSA.1 | | | | | X | | |
| FMT_SMF.1 | | | | | X | | |
| FCS_COP.1/DES | | | | | | X | |
| FCS_COP.1/3DES | | | | | | X | |
| FCS_COP.1/SHA | | | | | | X | |
| FDP_SDI.2 | X | | | | | | |

# 9 References

## 9.1 User Guidance

Table 20: User guidance

| | |
|---|---|
| [HardwareManual] | SLE 88CNFX Family – Hardware Reference User´s Manual; Infineon Technologies AG; Edition 2009-12-19 |
| [SoftwareManual] | SLE 88 Family SLE88CNFXxxxxPM PSL & Security Reference Manual; Infineon Technologies AG; Edition 2011-05 |
| [ErrataSheet] | SLE88CNFX Family Errata Sheet; Edition 2010-07-02 |

## 9.2 Literature

Table 21: Rules and standards

| | | |
|---|---|---|
| [1] | Security IC Platform Protection Profile BSI-PP-0035 | Version 1.0 15.06.2007 |
| [Common Criteria]<br>[2]<br>[3]<br>[4] | Common Criteria for Information Technology Security Evaluation<br>Part 1: Introduction and general model CCMB-2009-07-001<br>Part 2: Security functional requirements CCMB-2009-07-002<br>Part 3: Security Assurance Components CCMB-2009-07-003 | Version 3.1 Revision 3, 3. July 2009 |
| [11] | Joint Interpretation Library, Application of Attack Potential to Smartcards | Version 2.7 February 2009 |
| [AIS31] | Functionality classes and evaluation methodology for physical random number generators | AIS31, Version 1, 25.9.2001 |
| [ETSI TS 102 613] | Smart Cards; UICC-CLF interface; Physical and data link layer characteristic (Release 7) | V2.0.0 (2007-10) |

Note that the versions of these documents will be defined at the end of the evaluation and listed in the certification report.

## 9.3 List of abbreviations

| | |
|---|---|
| CC | Common Criteria |
| CI | Chip Identification mode (STS-CI) |
| CID | Chip Identification Data |
| CIM | Chip Identification Mode (STS-CI), same as CI |
| CPU | Central Processing Unit |
| CRC | Cyclic Redundancy Check |
| DPA | Differential Power Analysis |
| DFA | Differential Failure Analysis |
| ECC | Error Correction Control |
| EMA | Electro magnetic analysis |
| HW | Hardware |
| IC | Integrated Circuit |
| ID | Identification |
| I/O | Input/Output |
| SM | Security Mechanism |
| MED | Memory Encryption and Decryption |
| MMU | Memory Management Unit |
| NVM | Non Volatile Memory |
| O | Object |
| OS | Operating system |
| PLL | Phase Locked Loop |
| PSL | Platform Support Layer |
| RAM | Random Access Memory |
| RNG | Random Number Generator |
| ROM | Read Only Memory |
| S | Subject |
| SF | Security Feature |
| SFR | Security Functional Requirement |
| SFR | Special Function Register |
| SPA | Simple power analysis |
| STS | Self Test Software |
| SW | Software |
| T | Threat |
| TM | Test Mode (STS) |
| TOE | Target of Evaluation |
| UM | User Mode (STS) |

## 9.4 Glossary

| | |
|---|---|
| Application Program/Data | Software which implements the actual TOE functionality provided for the user or the data required for that purpose |
| Threat | Action or event that might prejudice security |
| Operating System | Software which implements the basic TOE actions necessary for operation |
| Central Processing Unit | Logic circuitry for digital information processing |
| Chip → Integrated Circuit | |
| Chip Identification Data | Data stored in the EEPROM containing the chip type, lot number (including the production site), die position on wafer and production week and data stored in the ROM containing the STS version number |
| Chip Identification Mode | Operational status phase of the TOE, in which actions for identifying the individual chip take place |
| Smart Card | Plastic card in credit card format with built-in chip |
| Controller | IC with integrated memory, CPU and peripheral devices |
| Cyclic Redundancy Check | Process for calculating checksums for error detection |
| End User | Person in contact with a TOE who makes use of its operational capability |
| Firmware | Part of the software implemented as hardware |
| Hardware | Physically present part of a functional system (item) |
| Integrated Circuit | Component comprising several electronic circuits implemented in a highly miniaturized device using semiconductor technology |
| IC dedicated software | Software used for testing purposes during production only but may also provide additional services to facilitate usage of the hardware and/or to provide additional services |
| Internal Random Access Memory | RAM integrated in the CPU |
| Non Volatile Memory (NVM) | Nonvolatile memory permitting electrical read and write operations |
| Security Mechanism | Logic or algorithm which implements a specific security function in hardware or software |
| Memory Encryption and Decryption | Method of encoding/decoding data transfer between CPU and memory |
| Microcontroller → Controller | |
| Microprocessor → CPU | |
| Object | Physical or non-physical part of a system which contains information and is acted upon by subjects |
| Random Access Memory | Volatile memory which permits write and read operations |
| Random Number Generator | Hardware part for generating random numbers |
| Read Only Memory | Nonvolatile memory which permits read operations only |

| | |
|---|---|
| Self Test Software | Part of the firmware with routines for controlling the operating state and testing the TOE hardware |
| Security Enforcing Function | Part(s) of the TOE used to implement part(s) of the security objectives |
| Security Target | Description of the intended state for countering threats |
| Software | Information (non-physical part of the system) which is required to implement functionality in conjunction with the hardware (program) |
| Memory | Hardware part containing digital information (binary data) |
| Subject | Entity, generally in the form of a person, who performs actions |
| Target of Evaluation | Product or system which is being subjected to an evaluation |
| Test Mode | Operational status phase of the TOE in which actions to test the TOE hardware take place |
| User Mode | Operational status phase of the TOE in which actions intended for the user takes place |

# 10 Appendix

The hash values listed in Table 22 are generated with the SHA-1 algorithm according the standard: *U.S. Department of Commerce, National Institute of Standards and Technology, Secure Hash Standard (SHA), FIPS PUB 180-3.*

Table 22: Reference hash values of the PSL V3.22.11

| SLE88CNFX6602PM, SLE88CNF6602PM, SLE88CNFX6602P, SLE88CNF6602P, SLE88CFX6602P, SLE88CF6602P | |
|---|---|
| Module | Hash Value |
| rompsl.ebf | f3361b3b 7ea4188c 302f7580 982fec16 c824fd39 |
| SLE88CNFX6600PM, SLE88CNFX5400PM, SLE88CNF6600PM, SLE88CNF5400PM, SLE88CNFX6600P, SLE88CNFX5400P, SLE88CNF6600P, SLE88CNF5400P, SLE88CFX6600P, SLE88CFX5400P, SLE88CF6600P, SLE88CF5400P | |
| Module | Hash Value |
| rompsl.ebf | 187f73dd 70b2f16f e6dfd865 5deee016 5dbd5838 |
| aes.obj | f254160f 7810b2de 1e9d870d c0c50bf8 7170f80c |
| aes128.obj | aec0ca88 09d33e73 9ca5ad28 2fb0531d fa7c39b0 |
| changetlbva.obj | fbd7aa5f 2cefdc0c 6d247acc 54dc971f 50490691 |
| crc.obj | ce292e81 229139b7 40806b48 c4c60a85 aeb9c2fb |
| crypto.obj | e21b4f75 0601f09f c0419ebd 49fbda12 055cbd85 |
| Crypto2kLib.obj | 62425a19 a67b6eba 4b38d8ed 8db1a162 84a25a0f |
| cryptocheckprime.obj | 95746ba1 54b98614 65e638a1 2d562f88 51024554 |
| cryptocreate1cand.obj | bfdbb217 50b4753e 6a013d50 0ae29f34 21ea146c |
| cryptocreateprime.obj | 24bbfdbf 4887aabd 340b46d2 9fa1846e 212f5b94 |
| cryptocreateprimeprod.obj | 6485561d ba4985c6 0426d72c c8b90c88 361f3346 |
| cryptodiv.obj | 662ceb2c 0536a5da 3a9121a7 8b545bb7 4663f245 |
| cryptodrv.obj | 52b8c339 e21fc9ec fe8b019a 1f8dcdfb 0db92418 |
| cryptoecgfpdrv.obj | 2cbe0d35 3f736165 424dca8a 51a37d7d 13043730 |
| cryptoecgfpmul.obj | ae5327dc bf390f0f a4b8ddc6 5c2be0d4 810985c0 |
| CryptoEcLib.obj | 0731306d 6cd5352f ef1c243d 053af437 4d90d00a |
| cryptoextensions.obj | 812a667b c653a1e7 6d18cf95 22459777 ffc2005d |
| cryptofnx.obj | c11e428d ebf8f98e e9edc906 bf39a01f 7ed73c3c |
| cryptofnxc.obj | 5231abff 5f8d9819 d0719120 d9391dc5 12f0569e |
| cryptomodinv.obj | 29594a51 e5680fdc 26c36166 383b3917 f48a0b7e |
| cryptomodmul2k.obj | 3125a498 1e2ba3fa d15833c1 06a16740 99ea25a7 |
| cryptomodpow.obj | 850f4113 ecab9d33 c3015209 2bacf718 1752394b |
| cryptomodpow2k.obj | fa28b69f 158b6774 d8e9d65f fd4253cc 16278372 |

| | |
|---|---|
| cryptomodpow_b3.obj | 7a6928df 2356296e 49607618 c6333ce4 29e56b27 |
| cryptonextprime.obj | 2b57f610 46deea90 1c3109c0 17a5f1e7 fb89a7ce |
| cryptorsacrtchk3_1k.obj | b2630ce3 280cdfe9 08bb872d a654ca29 3da98ee9 |
| cryptorsacrtchk3_2k.obj | 17ee8299 187f0ccb 55629b54 66011af6 edef173e |
| cryptorsacrtchk4.obj | 18d521f1 4255eb14 1b9faf53 a99fe70d 37666cff |
| cryptorsacrtgarner.obj | e0726a69 47ca8e74 3430332d 01f0f2db 1f5ffd06 |
| cryptorsasigncrt.obj | 35700f8f dac07cbf 9ace3f15 ca2823a7 b1b9212b |
| cryptorsasigncrt_m1.obj | c530d9dd de78b84e 381399d6 0a82bd32 2d97f6aa |
| cryptotransfer.obj | 1e3b1b2d 2f5338a7 e0f87624 916e7c32 1f6bdc28 |
| cryptotransmod2k.obj | e65c90a5 bc3e86eb fd294d73 a8bc1160 2327fad5 |
| cryptoutil.obj | 816e1c9a 3ef15907 054ac877 c28acde2 f7a87534 |
| cryptoutil2k.obj | 662d25aa a55f5fb2 bb5ab0aa 379ce967 0763fbd5 |
| ddes.obj | 43836341 01f97c38 60ba9594 e073dc10 83f7a988 |
| desextensions.obj | 22e5ef41 5d957d17 e4f5e1e2 fb784ed8 3ca5cc15 |
| eeprom.obj | ed313ed8 b018d249 91f7491a 1295cd36 6b6986f3 |
| fl2.obj | 9603aaa7 8ba070bb 630ca3fb c2dfd7dd 1e9f205e |
| fl2ar.obj | 8dad7904 8b8d76b8 3e9f4187 e382fe6b 2491bc75 |
| fl2keys.obj | 7aa63609 db9355e5 2a0088e8 fffa1e4d e660b32e |
| fl2mutualauth.obj | a13b53b7 cf710f32 bc17dd91 24fe99e0 ea1b77f1 |
| fl2PinLetter.obj | 0391244b 1cee411c 05da34d7 62432f75 74ac92d6 |
| fl2psa.obj | 4603b083 3184762a 0febcfa6 a1459cd3 207881af |
| fl2TrackingInfo.obj | 6765f339 bcd6b5da 32bf94a5 cad40b3d 7608cdd4 |
| ImmDriver.obj | 9716d2e7 bb38d487 09551c58 eab69e9e 2ddfe70d |
| loader.obj | c3e10686 4a3467b4 01b0caad 8a78c989 927896cb |
| loadershared.obj | 4c9917e6 e8ed2023 a08e0a61 f8e58609 2eacf22b |
| md5.obj | 3f9f391d fd0f68e8 a9660c2d 5ca465d1 900acc63 |
| mfmanagement.obj | 73fb9214 ac5d4085 1e2d3f32 c0b06cdd 5a917057 |
| mfreader.obj | d916b989 42c9c47a 5dff1981 5dc927ef 0d4d0110 |
| mmuAllocFast.obj | ab76317b e74f26a5 e799c8da 9f7bde6f f8f236bd |
| MmuAllocMem.obj | aa73f404 7813fa83 bd57f3bb e01a4e9a 09ad499b |
| MmuReadMem.obj | 515eafed 3a455611 c14f03b6 55504d24 d28c3e2f |
| MmuShareMem.obj | 017bba68 bc9e3f8d 84c3aa58 861315f4 a8f59580 |
| MmuSupportFunctions.obj | abafb348 5a17637d 429a47e0 30684142 843e5fda |
| nodePages.obj | e9552796 b3bfd265 2b2a20a9 b5c645b9 b451a290 |
| nodepagesasm.obj | 10be50f3 7be9e8e2 f5f0058e 167a6d31 25d75f6d |

| | |
|---|---|
| omdeletereg.obj | ea33d761 73dc5e2f 64b7b9ab 6211b6f4 30fd92e7 |
| ommodifyreg.obj | ad6e9b53 05256357 2110952d 3ef5146d b005fc76 |
| omwritereg.obj | b4954e51 baa90901 dc1c1150 72482b75 67fbf957 |
| omwriteutils.obj | e829855d 8a7fea57 4dd296f4 be566624 cf458f2d |
| pagefile.obj | 816aff91 1fb71a00 f9322131 b52925ed f750b0c6 |
| pl.obj | 3b0fc534 77ea095c b43b1923 c37d6d70 7e3c99f3 |
| protocol.obj | eb7cb36c b449b41c 11cd372e 54a0948a 73252309 |
| prot_atr_pps.obj | bb23912f 08cedf4f cbb7b5e0 3f9b1df2 35456018 |
| prot_t0.obj | 9ebacd54 81d8cda7 9c1c7abb 5796099d d3c4c028 |
| prot_t1.obj | 2bd167e8 6924a297 65f737f5 08495e55 4665e35b |
| psllibversion.obj | 37c905dc 997309c2 98912dd1 dea21973 cd14b3d5 |
| sha1.obj | 557dcada 90359fbf 8791b885 245add00 2c89c52a |
| sha2.obj | d746f98c b4ef3a1c ffbfbc19 9defdd1b 42bb3491 |
| SwpDriver.obj | ca903468 90d6289f d330e0e 5 f7a48c60 c03ba4ab |
| taskmanager.obj | afdf0057 fe700bca ab7a3a55 c0892cd0 da1c53a0 |
| tasksasm.obj | 2584be7a 7dbee637 fdcf1834 26c76838 f87ce12e |
| ufl.obj | d39e12fd 44fd75e3 97dd533e f8c68557 311a05f9 |