Public

Infineon Technologies AG
Chip Card & Security IC´s

Evaluation Documentation

SLE88CFX4001P / m8835

Security Target

Version 1.6
Date 29-11-2006
Author Jürgen Noller

Print Date:	**06.12.2006 17:27**
Filename:	**SLE88CFX4000P_SecTar.doc**

Revision History

Version	Page	Subject
1.0		22-11-2005: PSL version corrected, SDK version added
1.1		20-01-2006: Basic PSL, B17 and Delta Manual for SDK2.9 SP4 added
1.2		03-04-2006: Changes for Re-Certification
1.3		25-04-2006: Changes at section 8.1 and 8.2
1.4		30-06-2006: Changes for B18 Re-certification
1.5		18-08-2006: Updates for CC Version 2.3
1.6		29-11-2006: Update of User Guidance

TABLE OF CONTENTS

List of figures:

List of tables:

1 Introduction

1.1 Security Target Identification

The Security Target has the revision 1.6 and is dated 29-11-2006.

The Target of Evaluation (TOE) a smart card IC (Security Controller) which is named SLE88CFX4001P, is internally registered under the development code m8835b18 and has the version number b18.

The Security Target is based on the Protection Profile [BSI_PP].

The Protection Profile and the Security Target are built with Common Criteria V2.1.

Table 1: Identification

	Version number	Date	Registration
Security Target	1.6	29-11-2006	
Target of Evaluation SLE88CFX4001P SLE88CFX4003P SLE88CFX3521P SLE88CFX2921P	b18		 m8835b18 with PSL V2.00.07 and guidance documentation m8837b18 with PSL V2.00.07 and guidance documentation m8857b18 with PSL V2.00.07 and guidance documentation m8859b18 with PSL V2.00.07 and guidance documentation
Guidance Documentation	Edition 2006-11 Edition 2006-07	November, 2006 July, 2006	SLE88 Family - SLE88CFXxxxxP PSL & Security Reference Manual SLE88 Family - Hardware Reference Manual SLE88CFXxxx1P/3P Errata Sheet
[BSI_PP]	1.0	July 2001	BSI-PP-0002
Common Criteria	2.3	August 2005	Common Criteria for Information Technology Security Evaluation

Remarks to the Target of Evaluation (TOE):

The TOE of this Security Target encloses the SLE88CFX4001P and three different chip derivates. The hardware and the firmware of the SLE88CFX4001P and the three derivates are identical (the version number is for all the b18 and the firmware version is for all the PSL V2.00.07. The differences between the derivates are the NVM and ROM size as shown in the following:

SLE88CFX4001P:	m8835b18	400 kByte NVM, 80 kByte ROM, 0 kByte User ROM
SLE88CFX4003P:	m8837b18	400 kByte NVM, 80 kByte ROM, 160 kByte User ROM
SLE88CFX3521P:	m8857b18	352 kByte NVM, 80 kByte ROM, 0 kByte User ROM
SLE88CFX2921P:	m8859b18	292 kByte NVM, 80 kByte ROM, 0 kByte User ROM

The TOE, called SLE88CFX4001P in the following description, stands for the SLE88CFX4001P and the three derivates SLE88CFX4003P, SLE88CFX3521P and SLE88CFX2921P.

The firmware version PSL V2.00.07 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 SLE88CFX4003P. For the derivates SLE88CFX4001P, SLE88CFX3521P and SLE88CFX2921P 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 AES functionality, which is a part of SEF9 according to P.Add-Functions. Not including the code implementing the AES 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 AES 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.

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.

The TOE can be configured with activated or deactivated Supply Shutdown Mode (SSM). The configuration is done during the manufacturing process of the TOE according to the choice of the user.

1.2 Security Target Overview

The Target of Evaluation (TOE), the SLE88CFX4001P chip, is a smart card IC (Security Controller) meeting the highest requirements in terms of performance and security. It is manufactured by Infineon Technologies AG in a 0,13 µm CMOS technology. The IC is intended to be used in smart cards for particularly security-relevant applications.

In this security target the TOE (target of evaluation) 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 as the objectives of the security policy are defined as well as the security requirements. The requirements are built up of the security functional requirements as part of the security policy and the security assurance requirements as the steps during the evaluation and certification to show the TOE meets its requirements. The functionality of the TOE to meet the requirements is described.

The assets, threats, security objectives and the security functional requirements are defined in the Smartcard Integrated Circuit Platform Protection Profile [BSI_PP] and are referenced. These requirements build up a minimal standard common for all Smartcards.

The security enforcing functions are defined here in the security target as property of this specific TOE, the SLE88CFX4001P. Here it is shown how this specific TOE fulfils the requirements for the standard defined in the Protection Profile.

1.3 Conformance Claim

This security target claims conformance to

- Common Criteria for Information Technology Security Evaluation, Part1: Introduction and general model, August 2005, Version 2.3, CCMB-2005-08-01
- Common Criteria for Information Technology Security Evaluation, Part2: Security functional requirements, August 2005, Version 2.3, CCMB-2005-08-02
- Common Criteria for Information Technology Security Evaluation, Part3: Security assurance requirements, August 2005, Version 2.3, CCMB-2005-08-03
- Smartcard IC Platform Protection Profile, BSI-PP-002, [BSI_PP]

as follows

- Part 2 extended,
- Part 3 conformant,
- Package conformant to EAL5 augmented with components ALC_DVS.2, AVA_MSU.3 and AVA_VLA.4.

1.4 Security functional requirements and Augmentations

The security requirements of the TOE according to the [BSI_PP] are listed in Table 9. The augmented security functional requirements (see Table 10) are listed and described in section 5.1.

2 Description of the Target of Evaluation (TOE)

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 the [BSI_PP] as it belongs to the specific TOE.

2.1 Product Type

The Target of Evaluation (TOE), the SLE88CFX4001P chip, is a smart card IC (Security Controller) meeting the highest requirements in terms of performance and security. It is manufactured by Infineon Technologies AG 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 [BSI_PP]).

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 SLE88CFX4001P, whose block diagram is shown in Figure 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. A RNG (**R**andom **N**umber **G**enerator) is integrated on the chip.

The 32bit CPU is especially designed for smart card applications and provides powerful instructions for smart card applications. The memory comprises 16 kBytes of RAM (RAM), 80 kBytes of user ROM and 400 kBytes of NVM. 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 allow to operate the SLE88CFX4001P with a multiplication factor over the external clock signal or free running with maximum frequency. The RNG does not supply a pseudorandom number sequence, but instead produces genuine random numbers meeting high demands.

Five modules for cryptographic operations are implemented on the TOE. The coprocessor Crypto@1408BIT is used for calculation of asymmetric algorithms like RSA. 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 modules AES (Advanced Encryption Standard), MD5 and SHA-1 (Secure Hash Algorithm) are included as software modules in the PSL. The AES module is designed to counter attacks like DPA or EMA.
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, RNG (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-1 (Secure Hash Algorithm). The PSL provides the user (operating system) with the functionality to load code and data to 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 Infineon Technologies AG.

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 Infineon Technologies AG the STS provides an interface to the test capabilities of the SLE88CFX4001P. 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 optimised power consumption at minimal chip size. It therefore constitutes the basis for future smart card applications.

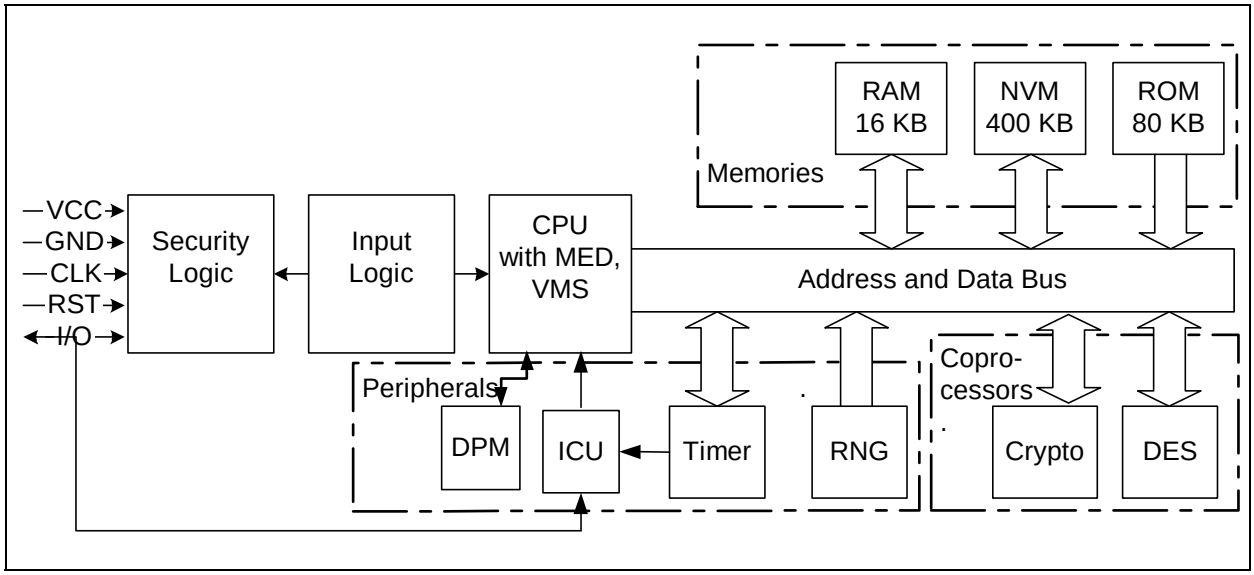


Figure 1: Block diagram of the SLE88CFX4001P hardware components

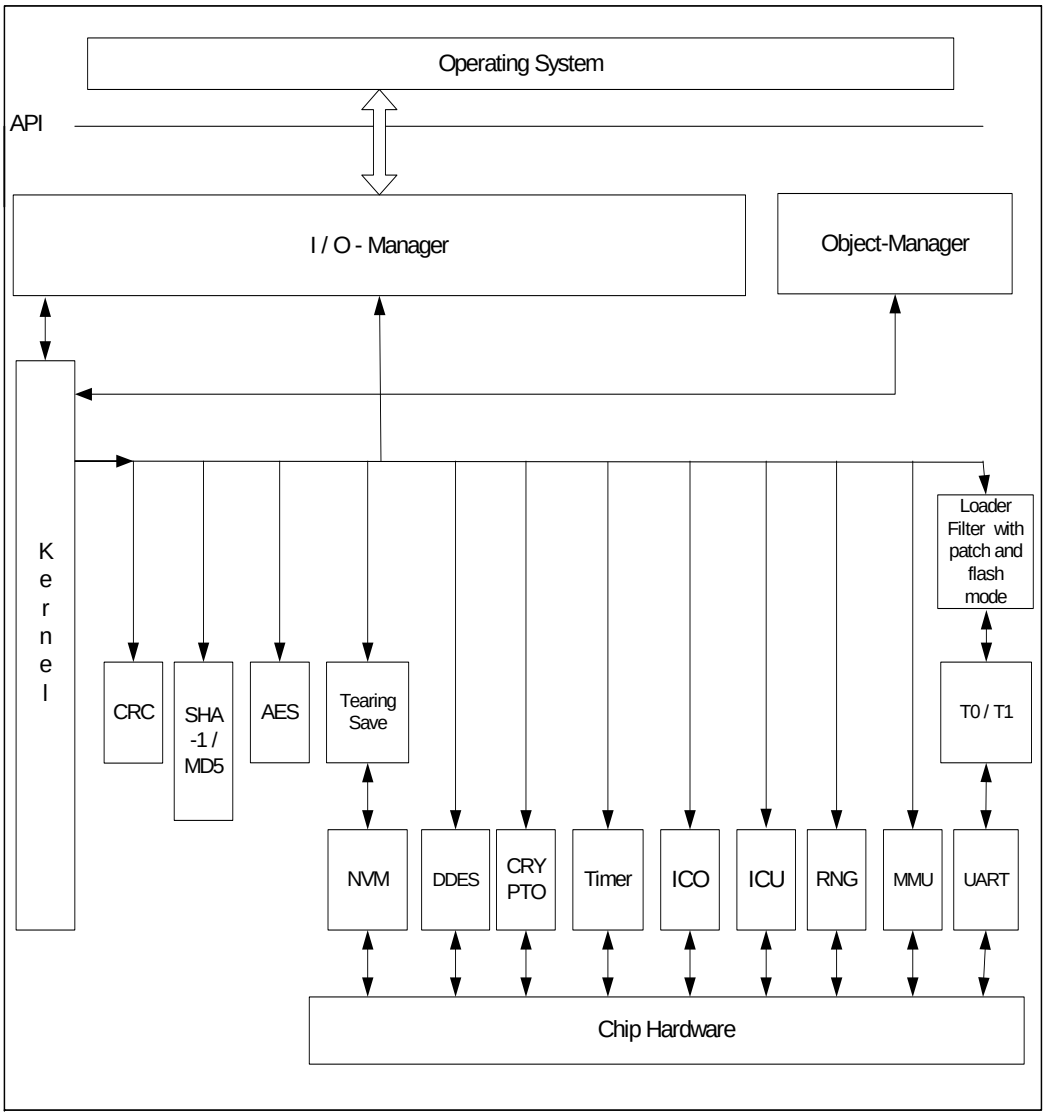


Figure 2: Block diagram of the SLE88CFX4001P Platform Support Layer (PSL)

2.2 Scope of the TOE

The TOE comprises the *hardware* of the smart card security controller, type SLE88CFX4001P, manufactured by Infineon Technologies AG, 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 Annex 9.1 are supplied as a manual. In the following description, the term “manufacturer” is short Infineon Technologies AG, the manufacturer of the TOE. The user software is not part of the TOE.

2.2.1 Hardware of the TOE

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

- Security logic (SEC)
- 32bit CPU with the subcomponents Memory Encryption and Decryption unit (MED) and Virtual Memory System (VMS)
- Peripheral modules comprising:
  - True random number generator (RNG)
  - Interrupt module (ICU)
  - Timer (TIM)
  - Internal oscillator (ICO)
  - Voltage regulator (VREG)
  - Universal Asynchronous Receiver Transmitter (UART)
  - Dynamic Power management
- External memory comprising:
  - 16 kBytes extended RAM
  - 80 kBytes ROM, including the test routines (STS) and the PSL
  - 400 kBytes 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
- Address and data bus (BUS)

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 [BSI_PP]). The other is the STS that consist of test and initialization routines (**S**elf **T**est **S**oftware, IC Dedicated Test Software in [BSI_PP]). 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) as defined in [BSI_PP] is comprised of:

- IO-Manager
- Kernel
- Object-Manager
- MD5 Generator Driver
- CRC Generator Driver (CRC)
- Secure Hash Algorithm Driver (SHA-1)
- AES Encryption/Decryption Driver (AES)
- NVM Driver (NVM)
- Tearing Save
- DDES Accelerator Driver (DDES)
- CRYPTO (Crypto@1408BIT)
- 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

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 RES, I/O0, I/O1, CLK lines and supply lines VCC and GND.
- The data-oriented I/O interface to the TOE is formed by the I/O pads (I/O0, I/O1).
- 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).

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 SLE88CFX4001P 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 SLE88CFX4001P. These recommendations are also included in the ordinary documentation.

The list of guidance documentation is given in Annex 9.1.

2.2.5 Forms of delivery

Several delivery processes exist during the lifecycle of the SLE88CFX4001P. 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 SLE88CFX4001P can be delivered in form of complete modules, in form of plain wafers or in an IC case (e.g. DSO20). Additionally the SLE88CFX4001P (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 [BSI_PP]. 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 may be produced in different production sites (listed in Table 2). The chip layout is not changed in this case and also the production testing does not differ. To distinguish the 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
Altis	5

3 TOE Security Environment

For this section the [BSI_PP] can be applied completely. A summary is given in the following.

3.1 Definition of Assets

The primary assets concern the User Data, which includes the data as well as program code (Smartcard Embedded Software). This asset has to be protected while being executed and on the other hand when the TOE is not in operation. This leads to the three primary assets

- User Data
- Smartcard Embedded Software
- TOE's correct operation

The specific functions of the TOE introduce additional assets.

- The random numbers generated by the TOE

The class of secondary assets consists of the following.

- Logical design data,
- Physical design data,
- IC Dedicated Software, Initialisation Data and Pre-personalisation Data, TSF data
- Specific development aids,
- Test and characterisation related data,
- Material for software development support, and
- Photomasks and products in any form

For details see [BSI_PP] section 3.1.

3.2 Assumptions

The assumptions defined in the [BSI_PP] concern the phases where the TOE has left the chip manufacturer.

A.Process-Card	Protection during Packaging, Finishing and Personalisation
A.Plat-Appl	Usage of Hardware Platform
A.Resp-Appl	Treatment of User Data

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

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

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 and (ii) the processing of User Data including cryptographic keys.

For details see [BSI_PP] section 3.2.

3.3 Threats

The threats are directed against the assets. The threat is a general description of “What one wants to do” and might contain several specific attacks (“How one wants to do it”). 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 [BSI_PP].

Table 3: Threats to Smartcards according to the Protection Profile

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

Due to the 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). Any restrictions are defined by the security policy of the specific application context and must be implemented by the Smartcard Embedded Software.

Table 4: Additional threats due to TOE specific functions and augmentations

T.Mem-Access	Memory Access Violation

For details see [BSI_PP] section 3.2.

3.4 Organisational Security Policies

The SLE88CFX4001P has to be protected during the first phases of his lifecycle (phases 2-TOE delivery)[1]. Later on the TOE has to protect itself. The organisational security policy covers this aspect.

P.Process-TOE Protection during TOE Development and Production

See [BSI_PP] for a detailed description.

3.4.1 Augmented organisational security policy

Due to the augmentations of the [BSI_PP] 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),
- Rivest-Shamir-Adleman Cryptography (RSA),
- Advanced Encryption Standard (AES),
- Secure Hash Algorithm (SHA-1).

[1] The TOE can be delivered either after phase 3 or after phase 4.

4 Security objectives

For this section the [BSI_PP] can be applied completely. Only a short overview is given in the following.

4.1 Security objectives for the TOE

See [BSI_PP].

Table 5: Objectives for Smartcards according to the Protection Profile

O.Phys-Manipulation	Protection against Physical Manipulation
O.Phys-Probing	Protection against Physical Probing
O.Malfunction	Protection against Malfunction due to Environmental Stress
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 shall provide “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),*
- *Rivest-Shamir-Adleman Cryptography (RSA),*
- *Advanced Encryption Standard (AES),*
- *Secure Hash Algorithm (SHA-1),*

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 is controlled as required, for example, in a multi-application environment.

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

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

4.2 Security objectives for the environment

The detailed description of the environmental security objectives is given in the [BSI_PP]. The list of objectives is in Table 7.

Table 7: Security objectives for the environment

Phase	**Objective for environment**	**Description**
Phase 1		
	OE.Plat-Appl	Usage of Hardware Platform
	OE.Resp-Appl	Treatment of User Data
Phase 2 up to TOE delivery		
	OE.Process-TOE	Protection during TOE Development and Production
TOE delivery up to end of phase 6		
	OE.Process-Card	Protection during Packaging, Finishing and Personalisation

4.2.1 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)".

Regarding the area based access control this objective of the environment has to be clarified. For the separation of different applications the Smartcard Embedded Software (Operating System) may implement a memory management scheme based upon security mechanisms of the TOE.

4.2.2 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 area based access control this objective 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.

4.2.3 Clarification of "Protection during Packaging, Finishing and Personalisation (OE.Process-Card)"

The protection during packaging, finishing and personalisation includes also the personalisation process (flash loader mode of the loader filter component) and the personalisation data (TOE software components) during Phase 4, Phase 5 and Phase 6.

5 IT security requirements

For this section the [BSI_PP] can be applied completely.

5.1 TOE security requirements

See [BSI_PP].

5.1.1 TOE security functional requirements

The detailed description of the security functional requirements is given in the [BSI_PP]. These security functional requirements are listed in Table 9. The additional security functional requirements are listed in Table 10. The necessary assignments are given in section 7.2. The description of the additional security functional requirements is given in the following.

5.1.1.1 Subset TOE security testing (FPT_TST.2)

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.

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 to verify the integrity of TSF data and stored TSF executable code which might violate the security policy. Therefore, the security functional component **Subset TOE security testing (FPT_TST.2)** has been newly created. This component allows that particular parts of the security mechanisms and functions provided by the TOE are tested.

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.
FPT_TST.2.1	The TSF shall run a suite of self tests *at the request of the authorized user* to demonstrate the correct operation of the *environmental mechanisms.*[2] *And of the RNG with help of the live test.*
Dependencies:	FPT_AMT.1 Abstract machine testing

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

5.1.1.2 Memory access control

Usage of multiple applications in one smartcard often requires to separate code and data in order to prevent that one application can access code and/or data of another application. To support this the TOE provides area based Memory Access Control.

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)"** addresses 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. Examples for such attributes are "*the memory area where the software is executed from, the memory area where the access is performed to, special information or properties tied to the software, and/or the operation to be performed*" (refer to below). The corresponding permission control information is evaluated 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 user 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)":

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 control *execution* accesses of the *software packages Security Layer (SL) and Platform Support Layer (PSL) residing in memory areas* on *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*[3].

The memory model of the SLE88CFX4001P 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 8.

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

Table 8: Effective access rights (EAR) for data read/write operations,
MPA is the “Memory Protection Access Violation” trap

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

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.

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

Dependencies: FDP_ACF.1 Security attribute based access control

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.

FDP_ACF.1.1 The TSF shall enforce the *Memory Access Control Policy* to objects based on the following: *software packages (subjects), data including code stored in memories (objects), the logic memory area where the software is executed from and 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 authorise 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*.

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

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.

FMT_MSA.3.1 The TSF shall enforce the *Memory Access Control Policy* to provide *well defined* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow *”privileged” subjects* to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

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.

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 *software with “privileged” attribute*.

Dependencies: [FDP_ACC.1 Subset access control or

FDP_IFC.1 Subset information flow control]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of management functions

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

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions*: usage of the relevant PSL function calls*.

Dependencies: No dependencies

5.1.1.3 Support of cipher schemas

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. The dependencies will be discussed in Section 8.2.

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

- Data Encryption Standard (DES),
- Triple Data Encryption Standard (3DES),
- Rivest-Shamir-Adleman (RSA)[4],
- *Advanced Encryption Standard (AES),*
- *Secure Hash Algorithm (SHA-1).*

DES Operation

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

FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 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 Mode (CBC)* and with cryptographic key sizes *of 56 bit* that meet the following *standards*:

U.S. Department of Commerce / National Bureau of Standards
Data Encryption Standard (DES), FIPS PUB 46-3, 1999 October 25.

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

FMT_MSA.2 Secure security attributes

Triple-DES Operation

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

FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 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*

[4] The TOE provides basic routines for the RSA calculation. For a secure RSA implementation the user has to implement additional countermeasures as described in a specific application note.

Block Mode (CBC) and with cryptographic key sizes *of 112 bit or 168 bit,* that meet the following *standards*:

U.S. Department of Commerce / National Bureau of Standards Data Encryption Standard (DES), FIPS PUB 46-3, 1999 October 25, keying option 2.

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
FMT_MSA.2 Secure security attributes

Rivest-Shamir-Adleman (RSA) operation

The Modular Arithmetic Operation of the TOE shall meet the requirement “Cryptographic operation (FCS_COP.1)” as specified below.

FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform *basic functions for encryption and decryption* in accordance with a specified cryptographic algorithm *Rivest-Shamir-Adleman (RSA)* and cryptographic key sizes *512-1326 bit* that meet the following *standards*

As described in Applied Cryptography, Second Edition, Protocols, Algorithms, and Source Code in C, Bruce Schneier, Section 19.3 RSA (page 467).

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
FMT_MSA.2 Secure security attributes

AES Operation

The AES Operation of the TOE shall meet the requirement “Cryptographic operation (FCS_COP.1)” as specified below.

FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *Advanced Encryption Standard (AES)* and cryptographic key sizes of *128 bit* that meet the following standards:

U.S. Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory (ITL), Advanced Encryption Standard (AES), FIPS PUB 197

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
FMT_MSA.2 Secure security attributes

SHA-1 Operation

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

FCS_COP.1 Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform *the calculation of a hash value* in accordance with a specified cryptographic algorithm *Secure Hash Standard (SHA-1)* and cryptographic key sizes of *a 160-bit output* that meet the following standards:

U.S. Department of Commerce, National Institute of Standards and Technology, Secure Hash Standard (SHA-1), FIPS PUB 180-1

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
FMT_MSA.2 Secure security attributes

5.1.1.4 Overview

Table 9: Security functional requirements defined in [BSI_PP]

Security Functional Requirement	
FRU_FLT.2	“Limited fault tolerance“
FPT_FLS.1	“Failure with preservation of secure state”
FPT_SEP.1	“TSF domain separation”
FMT_LIM.1	“Limited capabilities”
FMT_LIM.2	“Limited availability”
FAU_SAS.1	“Audit storage”
FPT_PHP.3	“Resistance to physical attack”
FDP_ITT.1	“Basic internal transfer protection”
FDP_IFC.1	“Subset information flow control”
FPT_ITT.1	“Basic internal TSF data transfer protection”
FCS_RND.1	“Quality metric for random numbers”

Table 10: 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.3	“Static attribute initialization”
FMT_MSA.1	“Management of security attributes”
FMT_SMF.1	“Specification of Management functions”
FCS_COP.1	“Cryptographic support”

5.1.2 TOE security assurance requirements

The evaluation assurance level is EAL 5 augmented. In Table 11 the security assurance requirements are given. The increase of the assurance components compared to the [BSI_PP] is expressed with bold letters. The augmentation of the assurance components to level EAL5 is given in italic letters.

Table 11: Assurance components

Aspect			**Refinement is done**
Configuration Management	ACM_AUT.1	Partial CM automation	
	ACM_CAP.4	Generation support and acceptance procedures	in PP
	ACM_SCP.3	**Development tools CM coverage**	**in ST**
Delivery and Operation	ADO_DEL.2	Detection of modification	in PP
	ADO_IGS.1	Installation, generation, and start-up procedures	in PP
Development	**ADV_FSP.3**	**Semiformal functional specification**	**in ST**
	ADV_HLD.3	**Semiformal high-level design**	
	ADV_IMP.2	Implementation of the TSF	
	ADV_INT.1	**Modularity**	
	ADV_LLD.1	Descriptive low-level design	
	ADV_RCR.2	**Semiformal correspondence demonstration**	
	ADV_SPM.3	**Formal TOE security policy model**	
Guidance Documents	AGD_ADM.1	Administrator guidance	in PP
	AGD_USR.1	User guidance	in PP
Life cycle Support	*ALC_DVS.2*	*Sufficiency of security measures*	*in PP*
	ALC_LCD.2	**Standardised life-cycle model**	
	ALC_TAT.2	**Compliance with implementation standards**	
Tests	ATE_COV.2	Analysis of coverage	in PP
	ATE_DPT.2	**Testing: low-level design**	
	ATE_FUN.1	Functional testing	
	ATE_IND.2	Independent testing – sample	
Vulnerability Assessment	**AVA_CCA.1**	**Covert channel analysis**	
	AVA_MSU.3	*Validation of analysis*	
	AVA_SOF.1	Strength of TOE security function evaluation	
	AVA_VLA.4	*Highly resistant*	

5.1.3 Refinements

Some refinements are taken unchanged from the [BSI_PP]. In some cases a clarification is necessary. In Table 11 an overview is given where the refinement is done. Two refinements from the [BSI_PP] have to be discussed here in the Security Target, as the assurance level is increased.

CM scope (ACM_SCP)

The refinement from the [BSI_PP] can be applied even at the chosen assurance level EAL 5 augmented with ACM_SCP.3. The assurance package ACM_SCP.2 is extended to ACM_SCP.3 with aspects regarding the development tools. The refinement is not touched.

Refinement for CM scope (ACM_SCP)

The “TOE implementation representation” within the scope of the CM shall include at

least:

- Logical design data,
- Physical design data,
- IC Dedicated Software,
- Smartcard Embedded Software,
- Final physical design data necessary to produce the photomasks, and
- Photomasks.

Functional Specification (ADV_FSP)

The refinement from the [BSI_PP] can be applied even at the chosen assurance level EAL 5 augmented with ADV_FSP.3. The assurance package ADV_FSP.2 is extended to ADV_FSP.3 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 [BSI_PP].

5.2 Security requirements for the Environment

5.2.1 Security requirements for the IT Environment

See [BSI_PP].

5.2.2 Security Requirements for the Non-IT-Environment

In the following the security requirements for the Non-IT-Environment are defined. For the development of the Smartcard Embedded Software (in Phase 1) the requirement RE.Phase-1 is valid.

RE.Phase-1 Design and Implementation of the Smartcard Embedded Software

The developers shall design and implement the Smartcard Embedded Software in such way that it meets the requirements from the following documents: (i) hardware data sheet for the TOE, (ii) TOE application notes, and (iii) findings of the TOE evaluation reports relevant for the Smartcard Embedded Software.

The developers shall implement the Smartcard Embedded Software in a way that it protects security relevant User Data (especially cryptographic keys) as required by the security needs of the specific application context.

The responsible parties for the Phases 4-6 are required to support the security of the TOE by appropriate measures:

RE.Process-Card Protection during Packaging, Finishing and Personalisation

The Card Manufacturer (after TOE Delivery up to the end of Phase 6) shall use adequate security measures 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).

The Smartcard Embedded Software shall meet the requirements “Cipher Schemas (RE.Cipher)” as specified below.

RE.Cipher Cipher Schemas

The developers of Smartcard Embedded Software must not implement routines in a way which may compromise keys when the routines are executed as part of the Smartcard Embedded Software. Performing functions, which access cryptographic keys could allow an attacker to misuse these functions to gather information about the key which is used in the computation of the function.

Keys must be kept 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 not possible 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 an appropriate key management has to be realised in the environment.

6 TOE summary specification

The product overview is given in section 2.1. In the following the security functionality is described and the relation to the security functional requirements is shown.

The TOE is equipped with ten security enforcing functions to meet the security functional requirements. The functions are:

SEF1: Operating state checking

SEF2: Phase management with test mode lock-out

SEF3: Protection against snooping

SEF4: Data encryption and data disguising

SEF5: Random number generation

SEF6: TSF self test

SEF7: Notification of physical attack

SEF8: Virtual Memory System (VMS)

SEF9: Cryptographic support

SEF10: NVM tearing save write

The following description of the security enforcing functions is a complete representation of the TSF.

6.1 SEF1: Operating state checking

Correct function of the SLE88CFX4001P is only given in the specified range of the environmental operating parameters. To prevent an attack exploiting that circumstances 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[5]. The defined secure state causes the chip internal reset process. The FPT_FLS.1 “Failure with preservation of secure state“ requirement is satisfied.

The covered security functional requirements are FRU_FLT.2 “Limited fault tolerance” and FPT_FLS.1 “Failure with preservation of secure state“.

The SEF1 does not use probabilistic or permutational effects.

[5] The operating state checking SEF1 can only work when the TOE is running and can not prevent reverse engineering.

6.2 SEF2: 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 implemented in the SLE88CFX4001P 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 SLE88CFX4001P 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 SLE88CFX4001P 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.

The phase management is used to provide the separation between the security enforcing functions and the user software. FPT_SEP.1 “TSF domain separation“ 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. 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 (FPT_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. The public part of the used signing key is stored in the NVM. This function could be deactivated permanently by the user software.

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

The test mode lock-out and the patch loader mode of the loader filter driver uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF *high*. The flash loader mode of the loader filter driver uses no probabilistic or permutational effects and is deactivated after its use.

6.3 SEF3: Protection against snooping

Several mechanisms protect the SLE88CFX4001P 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 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 protection against snooping uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF *high*.

6.4 SEF4: Data encryption and data disguising

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

The memory contents of the SLE88CFX4001P 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 components AES and RSA are 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 SEF4 uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF *high*.

6.5 SEF5: Random number generation

Random data is essential for cryptography as well as for physical security mechanisms. The SLE88CFX4001P is equipped with a true random generator based on physical probabilistic 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. The covered security functional requirement is FCS_RND.1 “Quality metric for random numbers”.

The SEF5 uses probabilistic or permutational effects and has to be included in the AVA_SOF analysis with SOF *high*.

6.6 SEF6: TSF self test

The TSF of the SLE88CFX4001P 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 function is SEF5, SEF7 and only specific environmental mechanisms of SEF1.

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“.

The TSF self test does not use probabilistic or permutational effects.

6.7 SEF7: Notification of physical attack

The entire surface of the SLE88CFX4001P is protected with the active shield. Attacks over the surface are detected when the shield lines are cut or get contact.

The attempt to use an opened device will be detected. The covered security functional requirement is FPT_PHP.3. Especially manipulation and the usage of galvanic contacts to gain information on the chip or the data is covered of this security enforcing function.
The SEF7 “Notification of physical attack” does not use probabilistic or permutational effects.

6.8 SEF8: Virtual Memory System (VMS)

The VMS in the SLE88CFX4001P 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 secure layer SL and the PSL. 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 initialisation”, FMT_MSA.1 “Management of security attributes” and FMT_SMF.1 “Specification of Management functions”.

The SEF8 ”Virtual Memory System (VMS) ” does not use probabilistic or permutational effects.

6.9 SEF9: 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 encryption, a combination of software and hardware unit to support RSA cryptography and software units to support the Advanced Encryption Standard (AES) and the Secure Hash Algorithm (SHA-1).
As defined the cryptographic operations are provided by the TOE, the covered security functional requirement is FCS_COP.1.

The SEF9 does use probabilistic or permutational effects, but cryptographic algorithms for encryption and decryption as well as key length are excluded from the SOF assessment. The Secure Hash Algorithm (SHA-1) is not considered as cryptographic algorithm for encryption and decryption and have to be included in the AVA_SOF analysis with SOF *high*.

6.10 SEF10: NVM tearing save 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 save 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 save 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.

The SEF10 ”NVM tearing save write ” does not use probabilistic or permutational effects.

6.11 Mapping of Security Functional Requirements

The justification of the mapping between Security Functional Requirements and the Security Enforcing Functions is given in sections 6.1- 6.10. The results are shown in Table 12. The security functional requirements are addressed by one relating security enforcing function except the security functional requirements FPT_FLS.1, FPT_PHP.3 and FRU_FLT.2.

The security functional requirements FPT_FLS.1 and FRU_FLT.2 are covered mutually supportive from hardware SEFs and software SEFs. FPT_FLS.1 “Failure with preservation of secure state” and FRU_FLT.2 “Limited fault tolerance” are covered from the SEF1 regarding the hardware aspects by filtering the external signals or resetting the TOE and SEF10 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 SEF3 for the aspect of making the reverse engineering harder even if the TOE is out of operation and from SEF7 for the aspect of detecting the attempt to modify the TOE when the chip is running. The SEF3 and the SEF7 are mutually supportive to cover FPT_PHP.3.

Table 12: Mapping of SFR and SEF

	SEF 1	SEF 2	SEF 3	SEF 4	SEF 5	SEF 6	SEF 7	SEF 8	SEF 9	SEF10
FAU_SAS.1		X								
FCS_RND.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				X			
FPT_SEP.1		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									X	

6.12 Assurance Measures

In Table 13 the TOE specific assurance measures are listed. These measures fulfil the requirements from Table 11.

This Security Target is the first document in the course of an evaluation. The exact references (version numbers and date) of the documents are not final during the evaluation of the security target. To avoid an update of the security target at the end of the evaluation the exact references are listed in the configuration list (ACM_SCP.3) of the evaluation.

Table 13: Assurance measures

Assurance	Assurance	Document
Security Target	ASE	Security Target
Configuration management	ACM_AUT.1	Configuration management (ACM)
	ACM_CAP.4	
	ACM_SCP.3	Configuration management scope (ACM_SCP)
Delivery and operation	ADO_DEL.2	Delivery (ADO)
	ADO_IGS.1	
Development	ADV_FSP.3	Functional Specification (ADV_FSP.3)
	ADV_HLD.3	High Level Design (ADV_HLD.3)
	ADV_IMP.2	Implementation (ADV_IMP.2)
	ADV_INT.1	High Level Design (ADV_HLD.3)
	ADV_LLD.1	
		Low Level Design (ADV_LLD.1)
	ADV_RCR.2	Representation Correspondance (ADV_RCR.2)
	ADV_SPM.3	[LKW_Model]
Guidance documents	AGD_ADM.1	Documentation (AGD)
	AGD_USR.1	
Life cycle support	ALC_DVS.2	Life Cycle Support (ALC)
	ALC_LCD.2	
	ALC_TAT.2	
Tests	ATE_COV.2	Test Documentation (ATE)
	ATE_DPT.2	
	ATE_FUN.1	
	ATE_IND.2	
Vulnerability assessment	AVA_CCA.1	Vulnerability Assessment (AVA)
	AVA_MSU.3	
	AVA_SOF.1	
	AVA_VLA.4	

7 PP claims

7.1 PP reference

This security target is conformant to the [BSI_PP].

7.2 PP tailoring

The assignments and selections foreseen in the [BSI_PP] are given in the following.

7.2.1 FCS_RND

The random numbers are generated from SEF5. The quality level of the random numbers is defined with the class “P2 high” of the [AIS31].

FCS_RND.1	Quality metric for random numbers
FCS_RND.1.1	The TSF shall provide a mechanism to generate random numbers that meet *the class “P2 high” criteria specified in [AIS31].*

7.3 PP additions

Additional objectives and security functional requirements are explicitly mentioned in this security target.

8 Rationale

The rationale from the [BSI_PP] is used here and it is not changed. The augmentations are designed to be conform to the rationale of the [BSI_PP]. The necessary extensions to the [BSI_PP] rationale are given in the following.

8.1 Security Objectives Rationale

Assumption, Threat or Organisational Security Policy	Security Objective
P.Add-Functions	O.Add-Functions
A.Key-Function	OE.Plat-Appl 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.

The justification related to the security objective “Area based Memory Access Control (O.Mem-Access)” is as follows: Since O.Mem-Access requires the TOE to implement exactly the same specific security functionality as required by T.Mem-Access, the threat 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 [BSI_PP] a 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 [BSI_PP] 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.

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.

8.2 Security Requirements Rationale

8.2.1 Rationale for the security functional requirements

Cryptographic operation (FCS_COP.1)

Objective	TOE Security Functional Requirements	Security Requirements for the environment
O.Add-Functions	FCS_COP.1 „Cryptographic operation“	RE.Phase-1 “Design and Implementation of the Smartcard Embedded Software” with RE.Cipher
OE.Plat-Appl OE.Resp-Appl		RE.Cipher

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 is suitable to meet the security objective.

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 requirement RE.Phase-1 and more specific by the 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
- FMT_MSA.2 Secure security attributes

to be met by the environment.

The dependencies FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2 must be covered from the environment (the smartcard embedded software) and are addressed by the requirement RE.Cipher.

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. In this case RE.Cipher requires that these functions ensure that confidential data (User Data) can not be disclosed while they are just being processed by the Smartcard Embedded Software. Therefore, with respect to the Smartcard Embedded Software the issues addressed by the objectives just mentioned are addressed by the requirement RE.Cipher.

The usage of cryptographic algorithms requires to use appropriate keys. Otherwise they do not provide security. The requirement RE.Cipher addresses these specific issues since cryptographic keys and other data are provided by the Smartcard Embedded Software. RE.Cipher requires that

keys must be kept confidential. They must be unique with a very high probability, cryptographically strong etc. If keys are imported into the TOE (usually after TOE Delivery), it must be ensured that quality and confidentiality is maintained. Therefore, with respect to the environment the issues addressed (i) by the objectives just mentioned and (ii) implicitly by O.Add-Functions are addressed by the requirement RE.Cipher.

In this ST the objectives for the environment OE.Plat-Appl and OE.Resp-Appl have been clarified. The requirement for the environment Re.Cipher has been introduced to cover the objectives OE.Plat-Appl and OE.Resp-Appl (in addition to O.Add-Functions). The Smartcard Embedded Software defines the use of the cryptographic functions FCS_COP.1 provided by the TOE.

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.

Subset security testing (FPT_TST.2)

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.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 to verify the integrity of TSF data and stored TSF executable code which might violate the security policy.

The tested security enforcing functions are SEF5, SEF7 and only specific environmental mechanisms of SEF1.

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.

Memory Access Control Policy

Objective	**TOE Security Functional Requirements**	**Security Requirements for the environment**
O.Mem-Access	- FDP_ACC.1 “Subset access control” - FDP_ACF.1 “Security attribute based access control” - FMT_MSA.3 “Static attribute initialisation” - FMT_MSA.1 “Management of security attributes” - FMT_SMF.1 “Specification of Management Functions”	RE.Phase-1 “Design and Implementation of the Smartcard Embedded Software”

The justification related to the security objective “Area based Memory Access Control (O.Mem-Access)” is as follows:

The security functional requirement “Subset access control (FDP_ACC.1)” with the related Security Function Policy (SFP) “Memory Access Control Policy” exactly require to implement an area based memory access control as demanded by O.Mem-Access. Therefore, FDP_ACC.1 with its SFP is suitable to meet the security objective.

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 requirement RE.Phase-1.

The security functional requirement “Static attribute initialisation (FMT_MSA.3)” requires that the TOE provides default values for security attributes. These default values can be overwritten by any subject (software) provided that the necessary access is allowed what is further detailed in the security functional requirement “Management of security attributes (FMT_MSA.1)”: The ability to update the security attributes is restricted to privileged subject(s). These management functions ensure that the required access control can be realised using the functions provided by the TOE.

8.2.2 Dependencies of security functional requirements

Security Functional Requirement	Dependencies	Fulfilled by security requirements
FCS_COP.1	FCS_CKM.1	Yes (by the environment)
	FDP_ITC.1 (if not FCS_CKM.1) or FDP_ITC.2 (if not FCS_CKM.1) FCS_CKM.4 FMT_MSA.2	

The dependencies FCS_CKM.1, FCS_CKM.4 and FMT_MSA.2 must be covered from the environment (the smartcard embedded software) and are addressed by the requirement RE.Cipher.

Security Functional Requirement	Dependencies	Fulfilled by security requirements
FPT_TST.2	FPT_AMT.1	See discussion below

The following discussion demonstrates how the dependencies defined by Part 2 of the Common Criteria for the requirement FPT_TST.2 are satisfied. The dependency defined in the Common Criteria is Abstract machine testing (FPT_AMT.1).

Part 2 of the Common Criteria explains that „the term ‘underlying abstract machine‘ typically refers to the hardware components upon which the TSF has been implemented. However, the phrase can also be used to refer to an underlying, previously evaluated hardware and software combination behaving as a virtual machine upon which the TSF relies.“

The TOE is already a platform representing the lowest level in a Smartcard. There is no lower or “underlying abstract machine” used by the TOE which can be tested. There is no need to perform testing according to FPT_AMT.1 and the dependency in the requirement FPT_TST.2 is therefore considered to be satisfied.

Security Functional Requirement	Dependencies	Fulfilled by security requirements
FDP_ACC.1	FDP_ACF.1	Yes
FDP_ACF.1	FDP_ACC.1	Yes
	FMT_MSA.3	Yes
FMT_MSA.3	FMT_MSA.1	Yes
	FMT_SMR.1	See discussion below
FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1	Yes
	FMT_SMR.1	See discussion below
	FMT_SMF.1	Yes

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.

8.2.3 Rationale for the Assurance Requirements and the Strength of Function Level

The chosen assurance level EAL 5 augmented determines the assurance requirements. In Table 11 the different assurance levels are shown as well as the augmentations. The augmentations are not changed compared to the Protection Profile

The assurance level EAL5 and the augmentation with the requirements ALC_DVS.2, AVA_MSU.3, and AVA_VLA.4 were chosen in order to meet assurance expectations. An assurance level of EAL5 is required for this type of TOE since it is intended to defend against highly sophisticated attacks without a protected environment. This evaluation assurance level was selected since it provides even a formal evidence on the conducted vulnerability assessment. In order to provide a meaningful level of assurance that the TOE provides an adequate level of defense against such attacks, the evaluators have access to all information regarding the TOE including the low level design and source code.

The rationale for the strength of function level from the [BSI_PP] is used as the level is not changed.

8.3 Security Requirements are Mutually Supportive and Internally Consistent

In addition to the discussion in section 7.3 of the [BSI_PP] the security functional requirement FCS_COP.1 is introduced. 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 also protect the cryptographic algorithms implemented according to the security functional requirement FCS_COP.1. Therefore, these security functional requirements support the secure implementation and operation of FCS_COP.1.

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 also protect the self-test functions implemented according to the security functional requirement FPT_TST.2. Therefore, these security functional requirements support the secure implementation and operation of FPT_TST.2.

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 also protect the area based memory access control function implemented according to the security functional requirement described in the security functional requirement FDP_ACC.1 with reference to the Memory Access Control Policy and details given in FDP_ACF.1. Therefore, those security functional requirements support the secure implementation and operation of FDP_ACF.1 with its dependent security functional requirements.

9 References

9.1 Documents and User Guidance

Table 14: User guidance

[HardwareManual]	SLE 88 Family – Hardware Reference Manual; Infineon Technologies AG; Edition 2006-07-13
[SoftwareManual]	SLE 88 Family – SLE88CFX4000P PSL & Security Reference Manual; Infineon Technologies AG; Edition 2006-11
[ErrataSheet]	SLE88CFXxxx1P/3P Errata Sheet

9.2 Literature

Table 15: Rules and standards

[BSI_PP]	Smartcard IC Platform Protection Profile	BSI-PP-0002; Version 1.0, July 2001
[Augmentations]	Smartcard Integrated Circuit Platform Augmentations	Version 1.0, March 8, 2002
[Common Criteria]	Common Criteria for Information Technology Security Evaluation Part 1, CCMB-2005-08-01 Part 2, CCMB-2005-08-02 Part 3, CCMB-2005-08-03	Version 2.3, August 2005
[LKW_Model]	A Formal Security Model of the Infineon SLE 88 A Formal Security Model of the Infineon SLE 88 Smart Card Memory Management	
[AIS31]	Functionality classes and evaluation methodology for physical random number generators	AIS31, Version1, 25.9.2001

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
CSFR	Core Special Function Register
DPA	Differential Power Analysis

DFA	Differential Failure Analysis
EMA	Electro magnetic analysis
HW	Hardware
IC	Integrated Circuit
ID	Identification
I/O	Input/Output
M	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
SEF	Security enforcing function
SFR	Security Functional Requirement
SigG	Signature law, see [SigG]
SigV	Signature regulation, see [SigV]
SPA	Simple power analysis
STS	Self Test Software
SW	Software
SO	Security objective
T	Threat
TLB	Translation lookaside buffer
TM	Test Mode (STS)
TOE	Target of Evaluation
UM	User Mode (STS)
UMC	Production site in Taiwan
USLC	User sensor life control, sensor self test during runtime

9.4 Glossary

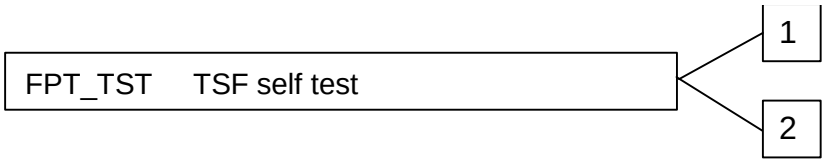
Application Program/Data	Software which implements the actual TOE functionality provided for the user or the data required for that purpose
Threat	Action or event that might prejudice security
Operating System	Software which implements the basic TOE actions necessary for operation
Central Processing 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

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 take place

10 Definition of the Security Functional Component FPT_TST.2

The following additions are made to „TSF self test (FPT_TST)“ in Common Criteria:

Component leveling



FPT_TST.1 TSF testing, provides the ability to test the TSF's correct operation. These tests may be performed at start-up, periodically, at the request of the authorised user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and executable code.

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 authorised user, or when other conditions are met. It also provides the ability to verify the integrity of TSF data and executable code.

The security functional component family “Subset TOE testing (FPT_TST.2)” is specified as follows.

FPT_TST.2	Subset TOE testing
Hierarchical to:	No other components.
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 authorised 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].

Dependencies: FPT_AMT.1 Abstract machine testing

11 Appendix

Table 16: Reference hash values of the PSL V2.00.07

SLE88CFX4001P, SLE88CFX3521P, SLE88CFX2921P	
Module	Hash Value
rompsl.ebf	fccca657 a5798038 b34723f3 982e309e ca25ae34
SLE88CFX4003P	
Module	Hash Value
rompsl.ebf	dfb23f2f 29806e22 fdb8e9c8 fbe27a0f 5fbe6a72
aes.obj	d8559e44 6d52a4fc d8eed005 ebf43506 ad802dc4
aes128.obj	45589af4 0820575d d234ae19 e30e9a9e a3746698
crc.obj	e78090c5 d81ccf75 b337b578 02d22df7 82f53d2d
crypto2klib.obj	31ca696d dbd897f6 84db8211 79e35ba7 5506ad3a
cryptocreate1cand.obj	55a920c2 44ef9708 efa7795e 771ed0f9 9caaef98
cryptocreateprime.obj	2c5baf66 23dfeca3 4b672395 f648b51d d654b739
cryptocreateprimeprod.obj	f48ccf1d 4bb2d327 78d700ac 6a91fb82 50d2bae4
cryptoecgfpdrv.obj	a93f3d57 d2ed33b0 2d8db354 446dad60 adbb3219
cryptoecgfpmul.obj	9d577590 852d6226 98a238cd e198cae5 5c4ec134
cryptoeclib.obj	59568451 c4ac82d9 53a26ada fc9250ae dc7fb1f4
cryptoextensions.obj	f0e8181a 28caf254 88817404 4113b84c fe43a1f8
cryptofnx.obj	d6e2b73a c02e991b 5b2cd887 c58aa96f bf7c2717
cryptofnxc.obj	580b9266 401c0caa 21862f0d 2255fa7e 8c197197
cryptomodmul2k.obj	d827ad42 59ac9eb2 20ca6111 fa0f02cd 4131184e
cryptomodpow2k.obj	3012fd99 e0eabba9 075e2f2c 7fb3ecda 4bcc8f22
cryptomodpow_b3.obj	18027a2e f4017654 6b9a0961 7904de05 50b1ba3d
cryptorsacrtchk3_1k.obj	3e5b0e26 28af2706 39cc9deb b6b888fe 37ad4d37
cryptorsacrtchk3_2k.obj	0a1b7176 7c58f1d6 b45d063b 527f4109 fe5f9652
cryptorsacrtchk4.obj	c26ff073 ef91021e 80d67e68 359959ca cb6f6966
cryptorsacrtgarner.obj	e48f250a 1d530077 c5fa5dc9 80dc97c1 94d50496
cryptorsasigncrt.obj	1dadc876 e576e453 9f7791dc b2513437 f66a4df0
cryptorsasigncrt_m1.obj	92b83bf3 60a7f027 8070179a 9821384e 2859c665
cryptotransmod2k.obj	1852a033 a7f6a830 477aba5f d9336bd4 7e92e0a9
cryptoutil2k.obj	dbe34d20 8b655433 d882bea6 8738b832 eb7ab674
desextensions.obj	725da772 a54e60e1 9d16c609 b73f0703 49515c13
fl.obj	f319fb31 8ae4e981 3f0c257a 87dbcfd5 228e1d43
fl2.obj	0f3ba10a 5207ae9c 5539d3a4 79373ad6 38d2838f
fl2ar.obj	d2e7fe4d 590ab55e 261ef820 278d397c bce083c2
fl2keys.obj	f2c2574e 68c0663e ededa409 3077a038 bffd2030

fl2mutualauth.obj	e60cadab 4be01351 3a6a1595 6f788d63 e66e8365
fl2psa.obj	05ed2226 18a9b2d0 7dfe009a 7d3b39d1 811bd0f3
loader.obj	9db7ae7a 9a94d777 5fb627b8 59db4cff 485ebbba
loadershared.obj	29f58a92 3daf9fa3 82f0f62e 792a78fc e03384c0
md5.obj	c807b961 591b0096 5705a0d9 b33ba108 8bcc7039
mmuextensions.obj	bf14cfaa 03d5be44 038309ea 81adafc4 a4b7a362
pl.obj	22feefc9 ed39c22c c87e32f0 7cd67d5a 64fea4e4
protocol.obj	2725aeb8 9b765366 0f9eb9fd 712a915c 55d14f56
prot_atr_pps.obj	e907e9a4 60c3d0f0 31f2ac98 1811b87f 28606bd5
prot_t0.obj	044d9694 209fac3f 59cdb360 42554edd da0296f2
prot_t1.obj	849a9247 c57bba61 b729953f 8909c157 f45bfec1
psllibversion.obj	e4138979 caa57433 2dc5a4a6 97174744 6950d739
sha1.obj	167e2237 600ca362 61441e15 5f9e8017 4f2d7a5e
stm_t1.obj	f38f0f20 72165cb1 e90f97a4 201a6beb b696c373