NXP Secure Smart Card Controllers P5CD016V1D / P5CD021V1D / P5CD041V1D / P5Cx081V1D with DESFire EV1

Security Target Lite

Rev. 1.1 — 24 October 2011

Evaluation documentation

Public

BSI-DSZ-CC-0707

Document information

Info	Content
Keywords	CC, Security Target Lite Lite, P5CD081V1D with DESFire EV1, P5CD081
Abstract	The document at hand is the Security Target Lite of NXP Secure Smart Card Controllers P5CD016V1D / P5CD021V1D / P5CD041V1D / P5Cx081V1D with DESFire EV1, which is developed and provided by NXP Semiconductors, Business Unit Identification according to the Common Criteria for Information Technology Security Evaluation Version 3.1 at Evaluation Assurance Level 4 augmented.

Revision history

Rev	Date	Description
1.0	16. May 2011	Derived from Security Target
1.1	24. October 2011	Table 1 deliveries updated

Latest version is: Rev. 1.1 (24 October 2011)

Contact information

For additional information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

1. ST Introduction

This chapter is divided into the following sections: “ST reference”, “TOE reference”, “TOE overview” and “TOE Description”.

1.1 ST reference

“NXP Secure Smart Card Controllers P5CD016V1D / P5CD021V1D / P5CD041V1D / P5Cx081V1D with DESFire EV1 Security Target Lite, Rev. 1.1, NXP Semiconductors, 24 October 2011”.

1.2 TOE reference

The TOE is a hardware platform with IC Dedicated Software named NXP Secure Smart Card Controllers P5CD016V1D / P5CD021V1D / P5CD041V1D / P5Cx081V1D with DESFire EV1 in the following also called P5CD081V1D with DESFire EV1. The TOE can be delivered in different major configurations. These major configurations support different memory types and a different contactless interface. Therefore they have own names for unique identification. All major configurations are based on the same hardware platform and the same IC Dedicated Software. The details of the configuration are explained in section 1.4.1.

1.3 TOE overview

1.3.1 Usage and major security functionality of the TOE

The TOE is the IC hardware platform P5CD081V1D with DESFire EV1 with IC Dedicated Software. The hardware platform incorporates an 8-bit processing unit, volatile and non-volatile memories accessible via a Memory Management Unit, cryptographic coprocessors, other security components and two communication interfaces.

This IC Dedicated Software includes IC Dedicated Support Software and IC Dedicated Test Software. The IC Dedicated Test Software is disabled before the TOE is delivered. The IC Dedicated Support Software consisting of the DESFire EV1 Software and the Boot-ROM Software. The Boot-ROM Software is used during each start-up of the TOE. The DESFire EV1 Software provides a set of functions to manage various kinds of data files stored in the non-volatile EEPROM partition for the DESFire EV1 Software.

The hardware platform and the IC Dedicated Support Software enforce a separation between the DESFire EV1 Software and the Security IC Embedded Software provided by the customer.

The TOE includes a Data Sheet for the hardware platform, a functional specification for the functionality provided by the DESFire EV1 Software, a document describing the Instruction Set of the hardware platform and the Guidance Document providing a description regarding the secure configuration and usage of the TOE.

The security functionality of the TOE is designed to act as an integral part of a complete security system in order to strengthen the design as a whole. Several security mechanisms are completely implemented in and controlled by the TOE. Other security mechanisms allow for configuration or even require handling of the response of the TOE by the Security IC Embedded Software. With different CPU modes and a Memory Management Unit the TOE is intended to support multi-application projects.

The hardware platform comprises different kind of on-chip memories. The ROM comprises the Security IC Embedded Software as well as the IC Dedicated Software.

The non-volatile EEPROM can be used as data or program memory for the Security IC Embedded Software and comprises a partition to store the data of the IC Dedicated Support Software. It contains high reliability cells, which guarantee data integrity. This is perfect for applications requiring non-volatile data storage and important for the use as memory for native programs. The RAM provides temporary storage during the operation for the CPU and the arithmetic coprocessor. Several security mechanisms protect the content of all memories.

A TOE used for high security applications in the banking and finance market or in electronic commerce applications shall:

- maintain the integrity and the confidentiality of code and data stored in its memories and
- maintain the different CPU modes with the related capabilities for configuration and memory access and
- maintain the integrity, the correct operation and the confidentiality of security functionality provided by the TOE.

This is ensured by the construction of the TOE and its security functionality.

The hardware platform of the P5CD081V1D with DESFire EV1 provides:

- Functionality to calculate the Data Encryption Standard (Triple-DES) with up to three keys,
- Functionality to calculate the Advanced Encryption Standard (AES) with different key lengths,
- Support for large integer arithmetic operations like multiplication, addition and logical operations, which is suitable for public key cryptography and elliptic curve cryptography,
- A Random Number Generator,
- Memory management control,
- Cyclic redundancy check (CRC) calculation,
- ISO/IEC 7816 contact interface with UART,
- Contactless interface supporting MIFARE DESFire and ISO/IEC 14443 A.

The IC Dedicated Support Software of the P5CD081V1D with DESFire EV1 is split into the Boot-ROM Software and the DESFire EV1 Software.

The Boot-ROM Software controls the secure start-up of the hardware platform and the DESFire EV1 Software provides:

- A set of functions used to manage the various kinds of data files stored in the non-volatile EEPROM memory

A detailed list of the software functions provided by the DESFire EV1 Software can be found in section 1.4.2.2

In addition, several security mechanisms are implemented to ensure proper operation as well as integrity and confidentiality of stored data. For example, this includes security mechanisms for memory protection and sensors, which allow operation under specified conditions only.

Note: Large integer arithmetic operations are intended to be used for calculation of asymmetric cryptographic algorithms. Any asymmetric cryptographic algorithm utilizing the support for large integer arithmetic operations has to be implemented in Security IC Embedded Software. Thus, the support for large integer arithmetic operations itself does not provide security functionality like cryptographic support. The Security IC Embedded Software implementing an asymmetric cryptographic algorithm is not included in this evaluation. Nevertheless the support for large integer arithmetic operations is part of the Security IC and therefore a security relevant component of the TOE, that must resist to the attacks mentioned in this Security Target and that must operate correctly as specified in the data sheet. The same scope of evaluation is applied to the CRC calculation.

1.3.2 TOE type

The TOE is the NXP Secure Smart Card Controllers P5CD016V1D / P5CD021V1D / P5CD041V1D / P5Cx081V1D with DESFire EV1. The TOE is a hardware platform including the IC hardware, IC Designer/Manufacturer proprietary IC Dedicated Test Software and IC Dedicated Support Software.

The TOE is delivered as a sawn wafer, or as module or other packaged form.

1.3.3 Required non-TOE hardware/software/firmware

None

1.4 TOE Description

The TOE consists of the IC hardware, the IC Dedicated Software and the associated documentation.

1.4.1 Physical Scope of TOE

The hardware platform of the P5CD081V1D with DESFire EV1 is manufactured in an advanced CMOS process. A block diagram is depicted in Figure 1.

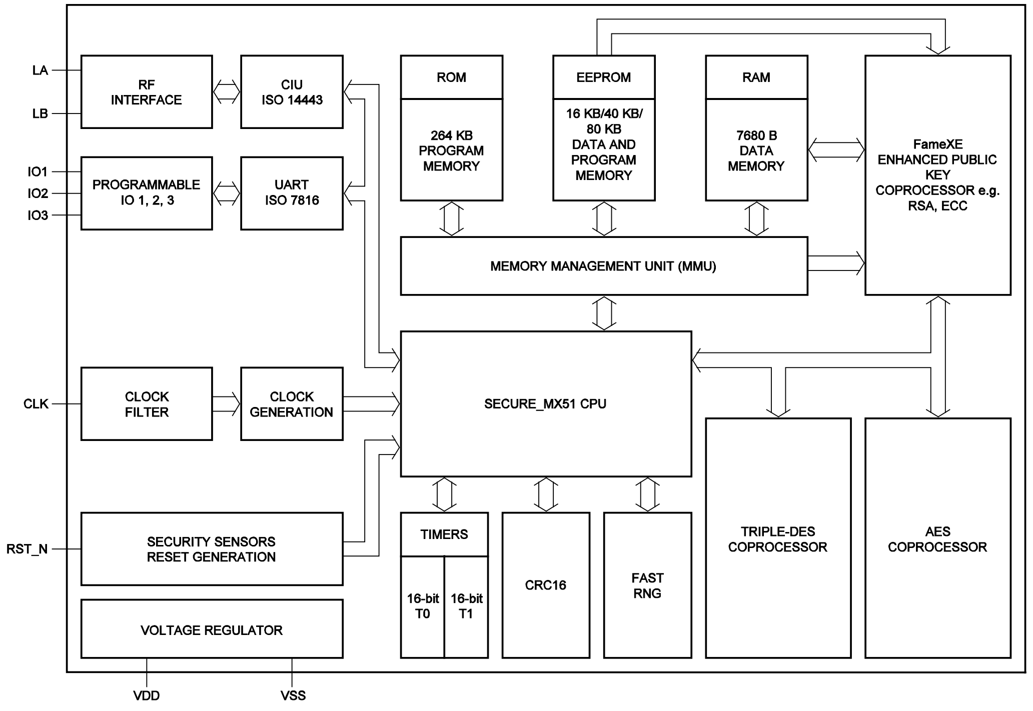

Figure 1 Block Diagram of P5CD081V1D

The IC Dedicated Software is a composition of IC Dedicated Test Software and IC Dedicated Support Software. The IC Dedicated Software is part of the ROM memory. All other software is called Security IC Embedded Software and is not part of the TOE. In the following Table 1 lists the TOE components.

TOE components

Table 1: Components of the TOE

Type	Name	Release	Date	Form of delivery
IC Hardware	NXP Secure Smart Card Controllers P5CD016V1D / P5CD021V1D / P5CD041V1D / P5Cx081V1D with DESFire EV1	V1D	T046B_20090210 .gds2 with changed metal 5 described in TO46D_ME5_20100128.gds2	Wafer, modules and packages (dice include reference T046D)
IC	Boot-ROM Software	1.3	11.07.2011	Test-ROM on the

Type	Name	Release	Date	Form of delivery
Dedicated Support Software				chip, TestRom_042_107. hex
	DESFire EV1 Software	1.3	11.07.2011	Test-ROM on the chip, TestRom_042_107. hex
IC Dedicated Test Software	Test-ROM Software	1.3	11.07.2011	Test-ROM on the chip, TestRom_042_107. hex
Document	Product data sheet P5CD016/021/041/051 and P5Cx081 family; Secure dual interface and contact PKI smart card controller [8]			Electronic document
Document	Data Sheet P5CD016/021/041 V1D and P5Cx081 V1D with MIFARE DESFire EV1 OS [18]			Electronic document
Document	Instruction Set, SmartMX-Family, Secure and PKI Smart Card Controller [9]			Electronic document
Document	MIFARE DESFire Functional Specification, NXP Semiconductors, Business Line Identification [11]			Electronic document
Document	Guidance, Delivery and Operation Manual NXP Secure Smart Card Controllers P5CD016V1D/P5CD021V1D/ P5CD041V1D/P5Cx081V1D [10]			Electronic document

The TOE supports different configurations. The configurations of the customer visible EEPROM size and the contactless interface are named major configurations because each configuration has its own product name. All major configurations support specific configuration options that must be addressed by the Security IC Embedded Software and may have an impact on the security of the TOE. Each major configuration provides further configurations, which are named minor configuration options.
The documentation covers all configurations

1.4.1.1 Evaluated major hardware configurations

The hardware platform can be configured to support different sizes of the EEPROM and different contactless interfaces. Five major configuration options are available, which are denoted by product names P5CD081V1D, P5CN081V1D, P5CD041V1D, P5CD021V1D and P5CD016V1D. All of them are equipped with a physical EEPROM of 80 kBytes and two interfaces comprising the ISO/IEC 7816 contact interface and the contactless interface. Their major differences are related to availability of EEPROM space and the behavior of contactless interface as detailed below.

The DESFire EV1 Software can be configured to support one of four possible configurations named A, D2, D4, and D8. The EEPROM memory sizes of these configurations are 2688 Byte for configuration D2, 5248 Byte for configuration D4, and 8320 Byte for configuration D8. Configuration A means that no EEPROM memory is reserved for DESFire and that the functionality of the DESFire EV1 Software is not available (except the CVEC2 call to Set/Get baud Rate). The EEPROM size available for the Security IC Embedded Software is reduced by the EEPROM size configured for the DESFire EV1 Software. The configuration of the DESFire EV1 Software is reflected in the commercial type names as listed in Table 3.

The minor configuration options of all major configurations are described in subsection 1.4.1.2.

Major configuration P5CD016V1D

P5CD016V1D supports all minor configuration options presented in subsection 1.4.1.2. Its major differences to other configurations are as follows.

- The EEPROM space available to the Security IC Embedded Software is reduced to 16 kBytes minus 256 Bytes, which are reserved for Security Rows (128 Bytes) and configuration data (128 Bytes).
- With the DESFire EV1 Software configuration A, D2 respectively D4 the EEPROM space available to the Security IC Embedded Software is limited to 16384 / 13696 / respectively 11136 Byte, both minus 256 Bytes manufacturer data as described above.
- The contactless interface is configured for contactless communication according to ISO/IEC 14443 A in [24] and [25].

Major configuration P5CD021V1D

P5CD021V1D supports all minor configuration options presented in subsection 1.4.1.2. Its major differences to other configurations are as follows.

- The EEPROM space available to the Security IC Embedded Software is reduced to 20 kBytes minus 256 Bytes, which are reserved for Security Rows (128 Bytes) and configuration data (128 Bytes).
- With the DESFire EV1 Software configuration A, D2 respectively D4 the EEPROM space available to the Security IC Embedded Software is limited to 20480 / 17792 / respectively 15232 Byte, both minus 256 Bytes manufacturer data as described above.
- The contactless interface is configured for contactless communication according to ISO/IEC 14443 A in [24] and [25].

Major configuration P5CD041V1D

P5CD041V1D supports all minor configuration options presented in subsection 1.4.1.2. Its major differences to other configurations are as follows.

- The EEPROM space available to the Security IC Embedded Software is reduced to 40 kBytes minus 256 Bytes, which are reserved for Security Rows (128 Bytes) and configuration data (128 Bytes).

- With the DESFire EV1 Software configuration A, D2 / D4 / D8 the EEPROM space available to the Security IC Embedded Software is limited to 40960 / 38272 / 35712 / 32640 Byte, all minus 256 Bytes manufacturer data as described above.
- The contactless interface is configured for contactless communication according to ISO/IEC 14443 A in [24] and [25].

Major configuration P5CD081V1D

P5CD081V1D supports all minor configuration options presented in subsection 1.4.1.2. Its major differences to other configurations are as follows.

- The physically implemented EEPROM of 80 kBytes is available to the Security IC Embedded Software except for 256 Bytes, which are reserved for Security Rows (128 Bytes) and configuration data of the manufacturer (128 Bytes).
- With the DESFire EV1 Software configuration A, D2 / D4 / D8 the EEPROM space available to the Security IC Embedded Software is limited to 81920 / 79232 / 76672 / 73600 Byte, all minus 256 Bytes manufacturer data as described above.
- The contactless interface is configured for contactless communication according to ISO14443 A in [24] and [25].

Major configuration P5CN081V1D

P5CN081V1D supports all minor configuration options presented in subsection 1.4.1.2. Its major differences to other configurations are as follows.

- The physically implemented EEPROM of 80 kBytes is available to the Security IC Embedded Software except for 256 Bytes, which are reserved for Security Rows (128 Bytes) and configuration data (128 Bytes).
- With the DESFire EV1 Software configuration A, D2 / D4 / D8 the EEPROM space available to the Security IC Embedded Software is limited to 81920 / 79232 / 76672 / 73600 Byte, all minus 256 Bytes manufacturer data as described above.
- The contactless interface is enabled and configured in S^2C mode. Therefore the TOE can be used for Near Field Communication (NFC) [26], for which an additional NFC helper IC is connected to IO3/SIGIN and LB/SIGOUT. However, it is possible that the TOE is powered in an appropriate electrical field when an antenna is connected to LA and LB. Pad I/O3 is connected to the NFC helper IC and cannot be used for contact communication according to ISO 7816.

1.4.1.2 Common minor configuration options

The following minor configuration options can be selected by the customer via order entry forms.

Table 2: Evaluated minor configuration options

Name	Values	Description
EDATASCALE	10h up to FFh	This value determines the size of the memory area available for the extended stack pointer. Refer to section 10.5 of [8].
Card Disable Function	Yes or No	When the Card Disable Function is enabled, the TOE can be locked completely. Once set by the Security IC Embedded Software, the execution of the Security IC Embedded Software is inhibited after the next reset. Refer to section 29.4 of [8].

Name	Values	Description
Block ROM read instructions executed from EEPROM	Yes or No	Instructions executed from EEPROM are allowed or not to read ROM contents. Refer to section 10.1.1.9 of [8].
128 Byte Page Mode	Yes or No	In the 128 Byte Page Mode, up to 128 Bytes of EEPROM can be programmed simultaneously, instead of up to 64 Bytes. Refer to section 10.9.1 of [8].
UID	Double	The size of the UID can be 7 bytes (Double UID). Refer to section 11.1.1 of [8].
Contactless communication protocol	(i) “proprietary protocol (compliant to ISO 14443 part 3)”; (ii) “T=CL protocol (compliant to ISO 14443 part 3 and part 4)”	Refer to section 21 of [8].
Maximum CIU Baudrate	106, 212, 424 or 848 kBaud	Defines the maximum available baudrate of the contactless interface. Refer to section 21 of [8].
Extended Voltage Class B activated	Yes or No	If Extended Voltage Class B is activated, the usable "3V supply voltage range" is extended to lower values than the minimum Class B supply voltage 2.7 V, and Class C operation is not supported. “Class BE” supply voltage range: 2.2 V ≤ V_{DD} ≤ 3.3 V. Refer to sections 34.1.3, 35.2, 29.2.2, 5 and 33 of [8].
Voltage Class C operation activated	Yes or No	If Voltage Class C is activated, supply voltage range is: 1.62 V ≤ V_{DD} ≤ 1.98 V. Refer to sections 34.1.3, 35.2, 29.2.2, 5 and 33 of [8].
Requested LA/LB input capacitance	17 pF or 69 pF	Additional capacitance (2x26 pF) between LA/LB required to meet resonance frequency at ID1/2 operation.

The values of all options listed in Table 2 can be chosen independently.

The Order Entry Forms [12], [13], [14], [15] and [16] contain a further option, which must be selected with a fixed value:

- The option “Allow execution from RAM” must be selected with “No”.

1.4.1.3 Evaluated package types

A number of package types are supported for each major configuration of the TOE. The commercial types are named according to the following format.

- P5CD016*pp*/T1D*rrffz* for major configuration P5CD016V1D
- P5CD021*pp*/T1D*rrffz* for major configuration P5CD021V1D
- P5CD041*pp*/T1D*rrffz* for major configuration P5CD041V1D
- P5CD081*pp*/T1D*rrffz* for major configuration P5CD081V1D
- P5CN081*pp*/T1D*rrffz* for major configuration P5CN081V1D

The commercial type name of each major configuration varies with the package type as indicated by the variable *pp*, - and with the Security IC Embedded Software as indicated by the variables *rr*, *ff* and *z*. The variables are replaced according to the rules in Table 3.

Table 3: Variable definitions for commercial type names

Variable	Definition
pp	This is a two character identifier for the package type, e.g. "UA" for a sawn wafer of 150µm thickness with electronically marked defects. The different types are defined in Table 4.
rr	ROM code number, different for every Security IC Embedded Software
ff	FabKey number, multiple keys are supported for each Security IC Embedded Software
z	MIFARE DESFire EV1 Configuration (F=A, B=D2, C=D4, D=D8)

Table 4 depicts the package types, which are supported in this Security Target, and assigns these to the major configuration types. The two characters in each entry of the table stand for *pp*, and identify the package type. An empty cell means that the Security Target does not support the respective package type for the corresponding major configuration.

Table 4: Supported commercial types

P5CD016V1D	P5CD021V1D	P5CD041V1D	P5CD081V1D	P5CN081V1D	
Ux	Ux	Ux	Ux	Ux	Wafer not thinner than 50µm (The letter "x" in "Ux" stands for a capital letter or a number, which identifies the waver type)
Xn	Xn	Xn	Xn		Module (The letter "n" in "Xn" stands for a capital letter or a number, which identifies the module type)
A4		A4	A4		MOB4 module
A6		A6	A6		MOB6 module
HN				HN	HVQFN32

For example, the commercial type name "P5CD081A4/T1Drrffz" denotes a P5CD081V1D in a MOB4 module and "P5CD081UA/T1Drrffz" denotes a P5CD081V1D on a 150µm sawn wafer inkless, which means that the defect ICs are electronically marked.

The package types do not influence the security functionality of the TOE. They only define which pads are connected in the package and for what purpose the chip can be used. Note that the security of the TOE is not dependent on which pad is connected or not – the connections just define how the product can be used. If the TOE is delivered as wafer the customer can choose the connection on his own.

Security during development and production is ensured for all package types listed above, please refer to section 1.4.3.

As already described above the complete resulting commercial type name is dependent on the customer software, i.e. the Security IC Embedded Software. Thus, a full commercial product name, which fits in the variable forms described in Table 4, determines an evaluated hardware, but gives no conclusion on the Security IC Embedded Software and whether it uses the proper hardware configuration as detailed in subsection 1.4.1.2.

1.4.2 Logical Scope of TOE

The TOE consists of the IC hardware, the IC Dedicated Software and the associated documentation. The following figure shows the logical boundary of the hardware platform with the IC Dedicated Software.

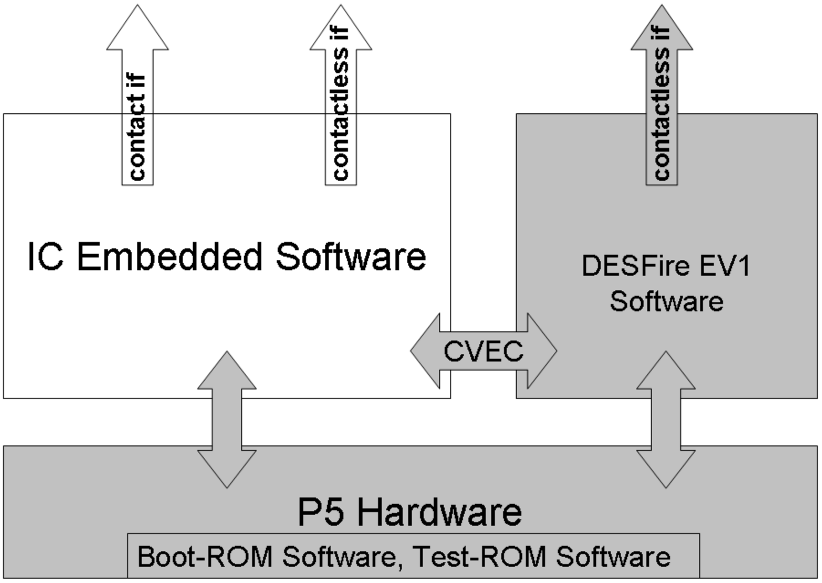

Figure 2 Logical boundary of the TOE

1.4.2.1 Hardware Description

The CPU of the P5CD081V1D with DESFire EV1 has an 8-bit architecture, with an 80C51 family instruction set extended by SMX specific instructions. The first and in some cases the second byte of an instruction are used for operation encoding. P5CD081V1D with DESFire EV1 distinguishes five CPU modes, which are summarized in the following table.

Table 5: CPU modes of the TOE

Super System Mode				
Boot Mode	Test Mode	MIFARE Mode	System Mode	User Mode

Boot Mode, Test Mode and MIFARE Mode are sub-modes of the so-called Super System Mode. These three modes cannot execute the Security IC Embedded Software. They are reserved for the IC Dedicated Software. The IC Dedicated Software is a composition of the Boot-ROM Software, the DESFire EV1 Software and the Test-ROM Software as introduced in section 1.4.1. The three software components are mapped one-to-one to the three CPU modes: In Boot Mode the TOE executes the Boot-ROM Software, in MIFARE Mode the TOE executes the DESFire EV1 Software and in Test Mode the TOE executes the Test-ROM Software. Please note that the Super System Mode is not a mode on its own: When the TOE is in Super System Mode, it is always either in Boot Mode, Test Mode or MIFARE Mode, depending on the settings of an internal register, which is not available to the Security IC Embedded Software.

The hardware platform of the P5CD081V1D with DESFire EV1 is able to control two different logical phases. After production of the IC hardware every start-up or reset completes with Test Mode and execution of the Test-ROM Software. The Test Mode is disabled at the end of the production test. Afterwards, every start-up or reset ends up in System Mode and execution of the Security IC Embedded Software.

System Mode and User Mode are available to the developer of the Security IC Embedded Software. The System Mode provides unlimited access to the hardware components. In User Mode the access is restricted to the CPU and specific Special Function Registers. Access rights to hardware components for software running in User Mode can be granted by software running in System Mode. The hardware components are controlled by the Security IC Embedded Software via Special Function Registers. Special Function Registers are interrelated to the activities of the CPU, the Memory Management Unit, interrupt control, I/O configuration, EEPROM, timers, UART, the contactless interface and the coprocessors.

Communication with P5CD081V1D with DESFire EV1 can be established via the contact interface through UART or direct usage of the I/O ports. Contactless communication can be established via the contactless interface unit (CIU) compatible to ISO/IEC 14443. If the P5CD081V1D with DESFire EV1 is configured to support the ISO/IEC 14443 interface and if the DESFire is configured as D2, D4 or D8 the communication layer support of the DESFire EV1 Software can be used by the Security IC Embedded Software to establish a contactless communication.

The P5CD081V1D with DESFire EV1 provides two types of interrupts: (i) exception interrupts, called “exception” in the following and (ii) event interrupts, called “interrupts” in the following. Exceptions and interrupts each force a jump to a specific fixed vector address in the ROM. Any exception and interrupt can therefore be controlled and guided by a specific part of the Security IC Embedded Software. In conjunction with the jump to a specific fixed vector address the IC hardware always enables a pre-defined CPU mode, which is either System Mode or User Mode. In addition, the P5CD081V1D with DESFire EV1 provides eight configuration vectors (CVEC) and 32 system call vectors (SVEC). These vectors have to be explicitly called by the Security IC Embedded Software. A jump to a configuration vector forces MIFARE Mode, a jump to a system call vector forces System Mode.

The switching between the different CPU modes and in particular the switching to Boot Mode and Test Mode are controlled with associated hardware mechanisms. These hardware mechanisms detect undefined transitions and enforce the separation between the CPU modes.

P5CD081V1D with DESFire EV1 incorporates 288 kBytes of ROM, 7680 Bytes of RAM and 80 kBytes of EEPROM. Access control to all three memory types is enforced by a

Memory Management Unit. The Memory Management Unit partitions each memory into two parts: The ROM is partitioned in 264 kBytes Application-ROM and 24 kBytes Test-ROM. The EEPROM is partitioned depending on the DESFire configuration. 256 Bytes of the EEPROM are always reserved for the manufacturer and either 2688 Byte, 5248 Byte or 8320 Byte of EEPROM are additionally reserved for the DESFire EV1 Software. The RAM is also partitioned, independent of the chosen configuration allocating 720 Bytes for the DESFire EV1 Software and the remaining 6960 Bytes for the application. Note that the ROM size is displayed as 264 kBytes in the block diagram in Figure 1 because only 264 kBytes are available to the Security IC Embedded Software.

In Test Mode the CPU has unrestricted access to the all memories. In Boot Mode and MIFARE Mode access is limited to the Test-ROM, the 256 Bytes EEPROM plus its configured part according to the configuration (A, D2, D4, D8) and the configured 720 Byte RAM partition of the DESFire EV1 Software. All other parts of the memories are accessible in System Mode and User Mode, namely the Application-ROM and the larger parts of EEPROM and RAM. User Mode is further restricted by the Memory Management Unit, which can be configured in System Mode.

The RAM, which is available to the Security IC Embedded Software, is further split in two parts. These are 4400 Bytes general purpose RAM and 2560 Bytes FameXE RAM. Both parts are accessible to the CPU, but the FameXE coprocessor can only access the FameXE RAM. The FameXE can access the FameXE RAM and the EEPROM without control of access rights by the Memory Management Unit. Since the Memory Management Unit does not control accesses of the FameXE, software which has access to the FameXE implicitly has access to the FameXE RAM. This also holds for the EEPROM, which is available to the Security IC Embedded Software. FameXE accesses to this part of the EEPROM are not controlled by the Memory Management Unit, i.e. software, which has access to the FameXE, implicitly has access to this part of the EEPROM. Due to the fact that software running in MIFARE Mode cannot communicate to the FameXE the FameXE can also not access this RAM and EEPROM areas. The reason for this is that the FameXE area and software running in MIFARE Mode are separated by the MIFARE Firewall.

The Triple-DES coprocessor supports single DES and Triple-DES operations. Only Triple-DES is used in this evaluation, in 2-key or 3-key operation. The AES coprocessor supports AES operation with three different key lengths. The FameXE coprocessor supplies basic arithmetic functions to support implementation of asymmetric cryptographic algorithms by the Security IC Embedded Software. The random generator provides true random numbers without pseudo random calculation.

P5CD081V1D with DESFire EV1 operates with a single power supply of 1.8 V, 3 V or 5 V nominal or within a field of a Proximity Coupling Devices. The maximum external clock frequency is 10 MHz nominal. P5CD081V1D with DESFire EV1 can be operated as well with an internal clock. P5CD081V1D with DESFire EV1 provides power saving modes with reduced activity. These are named IDLE Mode and SLEEP Mode, of which the latter one includes CLOCK STOP Mode.

The TOE protects secret data, which are stored to and operated by the TOE, against physical tampering. The security functionality of a Security IC is partially provided by the TOE, and completed by the Security IC Embedded Software. This causes dependencies between the security functionality of the TOE and the security functionality provided by the Security IC Embedded Software.

1.4.2.2 Software Description

Operating system and applications of a Security IC are developed by the customers. All software developed by the customer is summarized as Security IC Embedded Software. The Security IC Embedded Software is stored in the Application-ROM and/or in the EEPROM and is not part of the TOE. The Security IC Embedded Software depends on the usage of the Security IC.

The IC Dedicated Test Software, which is named Test-ROM Software, is stored to the Test-ROM and used by the manufacturer of the Security IC during production test. Before NXP Semiconductors delivers the product, the test functionality is disabled by disabling the Test Mode of the CPU. The IC Dedicated Test Software is developed by NXP and embedded in the Test-ROM. The Test-ROM Software includes the test operating system, test routines for the various design blocks of the hardware platform, control flags of life cycle status and configuration in the Security Row and shutdown functions to ensure that security relevant test operations cannot be executed illegally after phase 3.

The Dedicated Support Software is also stored to the Test-ROM and consists of two parts.

- The Boot-ROM Software, which is executed during start-up or reset of the TOE, i.e. each time when the TOE powers up or resets. It sets up the TOE and its basic configuration to ensure the defined initial values.
- The DESFire EV1 Software, which is started by a CVEC call of the Security IC Embedded Software. The DESFire EV1 Software provides the following functionality
  - Flexible file system that can contain up to 28 applications with up to 32 files in each application.
  - Support for different file types like values or data records.
  - Mutual three pass authentication, also according to ISO 7816-4.
  - Authentication on application level with fine-grained access conditions for files.
  - Multi-application support that allows distributed management of applications and ensures application segregation.
  - Data encryption for contact-less communication with replay attack protection.
  - Transaction system with rollback that ensures consistency for complex transactions.
  - Unique serial number for each device (UID) with optional random UID.

  In addition the software layer of the DESFire EV1 Software can be used to establish the contactless communication according to ISO/IEC 14443 for the Security IC Embedded Software. Please refer to [18] for more information.

1.4.2.3 Documentation

The data sheet “Product data sheet P5CD016/021/041/051 and P5Cx081 family; Secure dual interface and contact PKI smart card controller” [8] contains a functional description and guidelines for the use of the security functionality of the hardware platform, as needed to develop Security IC Embedded Software.

The data sheet addendum “Data Sheet P5CD016/021/041 V1D and P5Cx081 V1D with MIFARE DESFire EV1 OS” [18] is an addendum to the data sheet of the hardware platform which addresses the interfaces to DESFire EV1 Software, as needed to develop Security IC Embedded Software.

The functional specification of the DESFire EV1 Software is described in “MIFARE DESFire Functional Specification” [11]

The instruction set of the CPU is described in “Instruction Set, SmartMX-Family, Secure and PKI Smart Card Controller” [9].

The manual “Guidance, Delivery and Operation Manual NXP Secure Smart Card Controllers P5CD016V1D/P5CD021V1D/P5CD041V1D/P5Cx081V1D” [10] describes aspects of the program interface and the use of programming techniques to improve the security. The whole documentation shall be used by the developer to develop the Security IC Embedded Software. The documentation also includes guidance for the secure operation of the DESFire EV1 Software.

1.4.3 Security during Development and Production

The Security IC product life-cycle is scheduled in phases as introduced in the PP [6]. IC Development as well as IC Manufacturing and Testing, which are phases 2 and 3 of the life-cycle, are part of the evaluation. The Security IC is delivered at the end of phase 3 or after IC Packaging, which is at the end phase 4 in the life-cycle, and then part of the evaluation as well. The development and production environment of the TOE ranges from phase 2 to TOE Delivery.

With respect to Application Note 3 in [6] the TOE supports the authentic delivery using the FabKey feature or the originality key feature. For details on this features please refer to the data sheet in [8] and the manual in [10].

During the design and the layout process only people involved in the specific development project for an IC have access to sensitive data. Different people are responsible for the design data and for customer related data. The security measures installed within NXP ensure a secure computer system and provide appropriate equipment for the different development tasks.

The verified layout data is provided by the developers of NXP Semiconductors, Business Unit Identification directly to the wafer fab. The wafer fab generates and forwards the layout data related to the different photo masks to the manufacturer of the photo masks. The photo masks are generated off-site and verified against the design data of the development before the usage. Accountability and traceability are ensured among the wafer fab and the photo mask provider.

The production of the wafers includes two different steps regarding the production flow. In the first step the wafers are produced with the fixed masks independent of the customer. After that step the wafers are completed with the customer specific mask and the remaining fixed masks. The computer tracking ensures control of the complete process including storage of the semi-finished wafers.

The test process of every die is performed by a test centre of NXP. Delivery processes between the involved sites provide accountability and traceability of the produced wafers. NXP embeds the dice into modules, inlays or packages based on customer demand. Information about non-functional items is stored on magnetic/optical media enclosed with the delivery or the non-functional items are physically marked. In summary, the TOE can be delivered in different forms, which are

- dice on wafers
- smartcard modules on a module reel
- packaged devices in tubes or reels

The availability of major configuration options of the TOE in package types is detailed in section 1.4.1.3.

1.4.4 TOE Intended Usage

The end-consumer environment of the TOE is phase 7 of the Security IC product life-cycle as described in the PP [6]. In this phase the Security IC product provides the functionality defined by:

- the Security IC Embedded Software and if supported by the Security IC Embedded Software
- the IC Dedicated Support Software DESFire EV1 Software

The TOE is used by the end-consumer depending on the functionality implemented by the Security IC Embedded Software and the applications supported by the DESFire EV1 Software. Security ICs are used to assure authorized conditional access in a wide range of applications. Examples are identity cards, Banking Cards, Pay-TV, Portable communication SIM cards, Health cards, and Transportation cards. The end-user environment covers a wide spectrum of very different functions, thus making it difficult to monitor and avoid abuse of the TOE. The TOE is intended to be used in an insecure environment, which does not protect against threats.

The device is developed for most high-end safeguarded applications, and is designed for embedding into chip cards according to ISO/IEC 7816 [22] and for contactless applications. Usually a Security IC (e.g. a smartcard) is assigned to a single individual only, but it may also be used by multiple applications in a multi-provider environment. Therefore the TOE might store and process secrets of several systems, which must be protected from each other. The TOE then must meet security requirements for each single security module. Secret data shall be used as input for calculation of authentication data, calculation of signatures and encryption of data and keys.

The Security IC Embedded Software can call the DESFire EV1 Software that is part of the TOE. If the DESFire EV1 Software is called it provides its own security functionality and operates independent of the Security IC Embedded Software on the memory partitions assigned to the DESFire EV1 Software. The DESFire EV1 Software supports DESFire compatible applications in the field of:

- Electronic fare collection
- Stored value card systems
- Access control systems
- Loyalty

If privacy is an issue, the DESFire EV1 Software can be configured not to disclose any information to unauthorized users. However in this case also the application(s) implemented in the Security IC Embedded Software must support this privacy issue. Otherwise the privacy enforced by the DESFire EV1 Software can be circumvented by selecting another application of the TOE.

In development and production environment of the TOE the Security IC Embedded Software developer and system integrators such as the terminal software developer may use samples of the TOE for their testing purposes. It is not intended that they are able to change the behavior of the hardware platform in another way than an end-consumer.

The user environment of the TOE ranges from TOE delivery to phase 7 of the Security IC product life-cycle, and must be a controlled environment up to phase 6.

Note: The phases between TOE Delivery (end of phase 3 or phase 4) and phase 7 of the Security IC Product life-cycle are not part of the TOE construction process in the sense of this Security Target. Information about these phases is just included to describe how the TOE is used after its construction. Nevertheless such security functionality of the TOE, that is independent of the Security IC Embedded Software, is active at TOE Delivery and cannot be disabled by the Security IC Embedded Software in the following phases.

1.4.5 Interface of the TOE

The TOE supports two separate electrical interfaces.

- The contact interface comprises the pads connecting power supply, reset input, clock input, ground and the serial communication pads IO1, IO2 and IO3.
- The contactless interface comprises the two pads (called LA and LB) for the connection of an antenna.

Note: The S²C interface is based on a special configuration of the contactless interface and IO3.

The logical interface of the TOE depends on the CPU mode.

- In the Boot Mode the Boot-ROM Software is executed which provides no interface. There is no possibility to interact with this software.
- In the MIFARE Mode the DESFire EV1 Softwareis executed by the CPU. The interface of the DESFire EV1 Software comprises the command interface as defined by the functional specification of the DESFire EV1 Software, refer to [11].
- In System Mode and User Mode (used after TOE Delivery) the software interface is the set of instructions, the bits in the special function registers that are related to these modes and the physical address map of the CPU including memories. The access to the special function registers as well as to the memories depends on the CPU mode configured by the Security IC Embedded Software.

  Note: The logical interface of the TOE after TOE Delivery is based on the Security IC Embedded Software developed by the software developer. The Security IC Embedded Software can use both electrical interfaces. The identification and authentication of the user for the different CPU modes must be controlled by the Security IC Embedded Software.
- The Test Mode is disabled after production testing, before the TOE is delivered. Therefore no logical interface is visible in this CPU mode.

The chip surface can be seen as an interface of the TOE, too. This interface must be taken into account regarding environmental stress e.g. like temperature and in the case of an attack where the attacker manipulates the chip surface.

Note: An external supply and clock as well as a logical interface (terminal) are necessary for the operation of the TOE. Beyond the physical behavior the logical interface is defined by the Security IC Embedded Software or the DESFire EV1 Software as described above.

2. Conformance Claims

This chapter is divided into the following sections: “CC Conformance Claim", "Package claim", "PP claim", and "Conformance Claim Rationale”.

2.1 CC Conformance Claim

This Security Target claims to be conformant to version 3.1 of Common Criteria for Information Technology Security Evaluation according to

- “Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 3, July 2009, CCMB-2009-07-001” [1]
- “Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements, Version 3.1, Revision 3, July 2009, CCMB-2009-07-002” [2]
- “Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements, Version 3.1, Revision 3, July 2009, CCMB-2009-07-003” [3]

The following methodology will be used for the evaluation.

- “Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 3, July 2009, CCMB-2009-07-004” [4]

This Security Target claims to be CC Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in chapter 6.

2.2 Package claim

This Security Target claims conformance to the assurance package EAL 4 augmented. The augmentations to EAL4 are ALC_DVS.2 and AVA_VAN.5. In addition, the Security Target is augmented using the component ASE_TSS.2, which is chosen to include architectural information on the security functionality of the TOE.

Note: The PP “Security IC Platform Protection Profile” [6] to which this Security Target claims conformance (refer to section 2.3) requires also assurance level EAL4 augmented with the augmentation listed above.

The level of evaluation and the functionality of the TOE are chosen in order to allow the confirmation that the TOE is suitable for use within devices compliant with the German Digital Signature Law.

2.3 PP claim

This Security Target claims conformance to the Protection Profile (PP)

“Security IC Platform Protection Profile, Version 1.0, registered and certified by Bundesamt für Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0035” [6].

Since the Security Target claims conformance to this PP [6], the concepts are used in the same sense. For the definition of terms please refer to the PP [6]. These terms also apply to this Security Target.

The TOE provides additional functionality, which is not covered in the PP [6]. In accordance with Application Note 4 of [6] this additional functionality is added using the policies “P.Add-Components” and “P.DESFire-Emulation” (see section 3.3 of this Security Target).

2.4 Conformance Claim Rationale

According to section 2.3 this Security Target claims conformance to the PP “Security IC Platform Protection Profile” [6].

The TOE type defined in section 1.3.2 of this Security Target is a smartcard controller with IC Dedicated Support Software. This is consistent with the TOE definition for a Security IC with IC Dedicated Software provided in section 1.2.2 of [6].

All sections of this Security Target, where security problem definition, objectives and security requirements are defined, clearly state which of these items are taken from the PP [6] and which are added in this Security Target. Therefore this is not repeated here. Moreover, all additionally stated items in this Security Target do not contradict the items included from the PP (see the respective sections in this document). The operations done for the SFRs taken from the PP [6] are also clearly indicated.

The evaluation assurance level claimed for this target (EAL4+) is shown in section 6.2 which applies to the requirements claimed by the PP (EAL4+).

These considerations show that the Security Target correctly claims conformance to the PP [6].

3. Security Problem Definition

This Security Target claims conformance to the PP “Security IC Platform Protection Profile”, [6]. Assets, threats, assumptions and organizational security policies are taken from the PP [6]. This chapter lists these assets, threats, assumptions and organizational security policies, and describes extensions to these elements in detail.

The chapter is divided into the following sections: “Description of Assets", "Threats", "Organisational Security Policies", and "Assumptions".

3.1 Description of Assets

Since this Security Target claims conformance to the PP “Security IC Platform Protection Profile” [6] the assets defined in section 3.1 of [6] are applied here. These assets are cited here.

The assets related to standard functionality are:

- the User Data,
- the Security IC Embedded Software, stored and in operation,
- the security services provided by the TOE for the Security IC Embedded Software.

If the Security IC Embedded Software includes calls of the DESFire EV1 Software the assets are extended by:

- the Keys, Files and Values controlled by the DESFire EV1 Software
- the DESFire EV1 Software, stored and in operation.

Note that keys used by the Security IC Embedded Software for the cryptographic coprocessors are seen as User Data because the Security IC Embedded Software is not part of the TOE. Keys of the DESFire EV1 Software are explicitly addressed because their protection is completely provided by the TOE.

To be able to protect these assets the TOE shall protect its security functionality. To support this protection the critical information about the TOE design shall be protected. Critical information includes:

- logical design data, physical design data, IC Dedicated Software, configuration data,
- Initialization Data and Pre-personalization Data, specific development aids, test and characterization related data, material for software development support, photomasks.

3.2 Threats

Since this Security Target claims conformance to the PP “Security IC Platform Protection Profile” [6] the threats defined in section 3.2 of [6] are valid for this Security Target. The following table lists the threats defined in the PP [6].

Table 6: Threats defined by the PP [6].

Name	Title
T.Leak-Inherent	Inherent Information Leakage
T.Phys-Probing	Physical Probing
T.Malfunction	Malfunction due to Environmental Stress
T.Phys-Manipulation	Physical Manipulation

Name	Title
T.Leak-Forced	Forced Information Leakage
T.Abuse-Func	Abuse of Functionality
T.RND	Deficiency of Random Numbers

This Security Target defines additional threats related to the functionality provided by the DESFire EV1 Software. Considering Application Note 5 in [6] the following threats are defined by this ST:

T.Data-Modification Unauthorized modification of keys, files and values maintained by the DESFire EV1 Software.

Keys, files and values maintained by the DESFire EV1 Software are processed and stored by the TOE. They may be modified by unauthorized subjects. This threat applies to the processing of modified commands received by the TOE, it is not concerned with verification of authenticity.

T.Impersonate Impersonating authorized users during the authentication process of the DESFire EV1 Software.

An unauthorized subject may try to impersonate an authorized subject during the authentication sequence of the DESFire EV1 Software, e.g. by a man-in-the middle or replay attack.

T.Cloning Cloning using keys, files and values maintained by the DESFire EV1 Software

Keys, files and values maintained by the DESFire EV1 Software stored on the TOE may be read out by an unauthorized subject in order to create a duplicate.

3.3 Organisational Security Policies

Since this Security Target claims conformance to the PP “Security IC Platform Protection Profile” [6] the policy P.Process-TOE “Protection during TOE Development and Production” in [6] is applied here as well.

In accordance with Application Note 6 in [6] the following additional security policies are defined in this Security Target as detailed below.

The TOE provides specific security functionality, which can be used by the Security IC Embedded Software. In the following, specific security functionality is listed, which is not derived from threats identified for the TOE’s environment. It can only be decided in the context of the application against which threats the Security IC Embedded Software will use this specific security functionality.

The hardware platform shall provide the following additional security functionality to the Security IC Embedded Software “Additional Specific Security Components (P.Add-Components)” as specified below.

P.Add-Components Additional specific security components

- Triple-DES encryption and decryption
- AES encryption and decryption

- Area based Memory Access Control
- Memory separation for different software parts (including IC Dedicated Software and Security IC Embedded Software)
- Special Function Register Access Control.

In addition the DESFire EV1 Software as part of the hardware platform provides the following security functionality “P.DESFire-Emulation”. The Security IC Embedded Software can call the DESFire EV1 Software which implements this security policy. It is not mandatory for the Security IC Embedded Software to call the DESFire EV1 Software because the policy described above is independent of the DESFire EV1 Software. However if the TOE shall emulate the DESFire functionality the Security IC Embedded Software must call the DESFire EV1 Software. Therefore the IC Developer/Manufacturer defines the additional policies as specified below.

P.DESFire-Emulation The DESFire emulation provides the following specific security components:

- Confidentiality during communication provides the possibility to protect selected data elements from eavesdropping during contactless communication.
- Integrity during communication provides the possibility to protect the contactless communication from modification or injections. This includes especially the possibility to detect replay or man-in-the-middle attacks within a session.
- Transaction mechanism provides the possibility to combine a number of data modification operations in one transaction, so that either all operations or no operation at all is performed.

3.4 Assumptions

Since this Security Target claims conformance to the PP “Security IC Platform Protection Profile” [6] the assumptions defined in section 3.4 of [6] are valid for this Security Target. The following table lists these assumptions.

Table 7: Assumptions defined in the PP [6]

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

The following additional assumptions are added in this Security Target according to Application Notes 7 and 8 in [6].

A.Check-Init Check of pre-personalization data to ensure the authenticity of the TOE.

The Security IC Embedded Software must provide a function to check pre-personalization data and/or must support the use of the originality key function provided by the DESFire EV1 Software. The pre-personalization data is defined by the customer and the originality key is defined by NXP. Both data sets are injected by the TOE Manufacturer into the non-volatile memory to provide the possibility for TOE identification and for traceability.

The following additional assumption considers specialized encryption hardware of the TOE.

The developer of the Security IC 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 Security IC 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 Security IC 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.

The following two assumptions are outside the control of the DESFire EV1 Software. These assumptions must be implemented to support the security functionality of the DESFire EV1 Software.

A.Secure-Values — Usage of secure values

Only confidential and secure keys shall be used to set up the authentication and access rights for the DESFire EV1 Software. These values are generated outside the TOE. They must be protected during generation, management outside the TOE and downloaded to the TOE.

A.Terminal-Support — Terminal support to ensure integrity and confidentiality

The terminal verifies information sent by the TOE in order to ensure integrity and confidentiality of the communication.

Note that the assumptions A.Plat-Appl and A.Resp-Appl defined in the Protection Profile are relevant for all software running on the hardware platform. Therefore the assumptions are addressed during the development of the IC Dedicated Support Software. The Boot Software and the DESFire EV1 Software meet the assumptions A.Plat-Appl and A.Resp-Appl. The assumptions are still applicable for the Security IC Embedded Software and therefore they will remain in this Security Target as defined in the Protection Profile [6].

4. Security Objectives

This chapter contains the following sections: Security Objectives for the TOE, Security Objectives for the Security IC Embedded Software development Environment, Security Objectives for the Operational Environment and Security Objectives Rationale.

4.1 Security Objectives for the TOE

The TOE shall provide the following security objectives, which are taken from the PP "Security IC Platform Protection Profile" [6].

Table 8: Security objectives defined in the PP and the Hardware Security Target

Name	**Title**	**Defined in**
O.Leak-Inherent	Protection against Inherent Information Leakage	PP [6]
O.Phys-Probing	Protection against Physical Probing	PP [6]
O.Malfunction	Protection against Malfunctions	PP [6]
O.Phys-Manipulation	Protection against Physical Manipulation	PP [6]
O.Leak-Forced	Protection against Forced Information Leakage	PP [6]
O.Abuse-Func	Protection against Abuse of Functionality	PP [6]
O.Identification	TOE Identification	PP [6]
O.RND	Random Numbers	PP [6]

The Security Target defines additional security objectives. The following security objectives are provided by the hardware platform of the TOE:

O.HW-DES3 Triple DES Functionality

The TOE shall provide the cryptographic functionality to calculate a Triple DES encryption and decryption to the Security IC Embedded Software and the IC Dedicated Software. The TOE supports directly the calculation of Triple DES with up to three keys.

Note: The TOE will ensure the confidentiality of the User Data (and especially cryptographic keys) during Triple DES operation. This is supported by O.Leak-Inherent.

O.HW-AES AES Functionality

The TOE shall provide the cryptographic functionality to calculate an AES encryption and decryption to the Security IC Embedded Software and the IC Dedicated Software. The TOE supports directly the calculation of AES with three different key lengths.

Note: The TOE will ensure the confidentiality of the User Data (and especially cryptographic keys) during AES operation. This is supported by O.Leak-Inherent.

O.MF-FW MIFARE Firewall

The TOE shall provide separation between the DESFire EV1 Software as part of the IC Dedicated Support Software and the Security IC Embedded Software. The separation shall comprise software execution and data access.

O.MEM-ACCESS Area based Memory Access Control

Access by processor instructions to memory areas is controlled by the TOE. The TOE decides based on the CPU mode (Boot Mode, Test Mode, MIFARE Mode, System Mode or User Mode) and the configuration of the Memory Management Unit if the requested type of access to the memory area addressed by the operands in the instruction is allowed.

O.SFR-ACCESS Special Function Register Access Control

The TOE shall provide access control to the Special Function Registers depending on the purpose of the Special Function Register or based on permissions associated to the memory area from which the CPU is currently executing code. The access control is used to restrict access to hardware components of the TOE.

The possibility to define access permissions to specialized hardware components of the TOE shall be restricted to code running in System Mode.

The security objectives of the IC Dedicated Support Software (DESFire EV1 Software) can only be provided if this IC Dedicated Support Software is called by the Security IC Embedded Software. The DESFire EV1 Software is part of the TOE and provides the following security objectives:

O.DATA-ACCESS Access Control to DESFire Data

The TOE must provide an access control mechanism for data stored by the DESFire EV1 Software. The access control mechanism shall apply to read, modify, create and delete operations for data elements and to reading and modifying security attributes as well as authentication data. It shall be possible to limit the right to perform a specific operation to a specific user. The security attributes (keys) used for authentication shall never be output.

O.AUTHENTICATION Authentication

The DESFire EV1 Software as part of the TOE must provide an authentication mechanism in order to be able to authenticate authorized users. The authentication mechanism shall be limited to the DESFire EV1 Software and shall be resistant against replay and man-in-the-middle attacks.

O.CONFIDENTIALITY Confidential Communication

The TOE must be able to protect the communication of the DESFire EV1 Software by encryption. This shall be implemented by security attributes of the DESFire data element that enforce encrypted communication of the DESFire EV1 Software for the respective data element.

During DESFire operation the TOE shall also provide the possibility to detect replay or man-in-the-middle attacks within a session. This shall be implemented by checking verification data sent by the terminal and providing verification data to the terminal.

O.TYPE-CONSISTENCY Data type consistency

The TOE must provide a consistent handling of the data types (files and values) of the DESFire EV1 Software. This comprises over- and underflow checking for values, for data file sizes and for record handling.

O.TRANSACTION Transaction mechanism

The TOE must be able to provide a transaction mechanism that allows to update multiple data elements of the DESFire EV1 Software either all in common or none of them.

4.2 Security Objectives for the Security IC Embedded Software development Environment

In addition to the security objectives for the operational environment as required by CC Part 1 [1] the PP “Security IC Platform Protection Profile” [6] defines security objectives for the Security IC Embedded Software development environment which are listed below.

Table 9: Security objectives for the environment

Security objective	Description	Applies to phase...
OE.Plat-Appl	Usage of Hardware Platform	Phase 1
OE.Resp-Appl	Treatment of User Data	Phase 1

Clarification of “Usage of Hardware Platform (OE.Plat-Appl)”

The TOE supports cipher schemes as additional specific security functionality. If required the Security IC Embedded Software shall use these cryptographic services of the TOE and their interface as specified. When key-dependent functions implemented in the Security IC Embedded Software are just being executed, the Security IC 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)“.

If the Random Number Generator is used for leakage countermeasures, cryptographic operations (e.g. key generation) or cryptographic protocols (e.g. challenge response) these random numbers must be tested appropriately.

For multi-applications the Security IC Embedded Software (Operating System) can implement a memory management scheme based upon security functionality of the TOE to ensure the separation of applications.

Clarification of “Treatment of User Data (OE.Resp-Appl)”

By definition cipher or plain text data and cryptographic keys of the Security IC Embedded Software are User Data. The Security IC 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, if asymmetric algorithms are used, it must be ensured that it is not possible to derive the private key from a related public key using the attacks defined in this Security Target. 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 realized in the environment.

The treatment of User Data is also required when a multi-application operating system is implemented as part of the Security IC Embedded Software on the TOE. In this case the multi-application operating system will not disclose security relevant user data of one application to another application when it is processed or stored on the TOE.

4.3 Security Objectives for the Operational Environment

The following security objectives for the operational environment are specified according to the PP “Security IC Platform Protection Profile” [6].

Table 10: Security objectives for the operational environment, taken from the PP [6]

Security objective	Description	Applies to phase...
OE.Process-Sec-IC	Protection during composite product manufacturing	TOE delivery up to the end of phase 6

OE.Check-FabKey and/or OE.Check-OriginalityKey are defined to allow a TOE specific implementation (refer also to A.Check-Init). The security objective for the environment is split since the identification may be checked by different users.

OE.Check-FabKey — Check of pre-personalization data by the Security IC Embedded Software

To ensure the receipt of the correct TOE, the Security IC Embedded Software shall check a sufficient part of the pre-personalization data. This shall include at least the FabKey Data that is agreed between the customer and the TOE Manufacturer.

OE.Check-OriginalityKey — Check of the Originality Key of the DESFire EV1 Software

To ensure the receipt of the original DESFire Functionality, the Security IC Embedded Software shall support the use of the originality function provided by the DESFire EV1 Software. The originality key is introduced by NXP to check the authenticity of the DESFire product.

Note: It is not mandatory for the Security IC Embedded Software to use the DESFire EV1 Software or the originality key provided by NXP. For configuration A the originality function of the DESFire EV1 Software is not available.

OE.Secure-Values — Generation of secure values

The environment shall generate confidential and secure keys for authentication purpose of the DESFire EV1 Software. These values are generated outside the TOE and they are downloaded to the TOE during the personalization or usage in phase 5 to 7

OE.Terminal-Support Terminal support to ensure integrity and confidentiality

The terminal shall verify information sent by the DESFire EV1 Software in order to ensure integrity and confidentiality of the communication. This involves checking of MAC values, verification of redundancy information according to the cryptographic protocol and secure closing of the communication session.

4.4 Security Objectives Rationale

Section 4.4 in the PP “Security IC Platform Protection Profile” [6] provides a rationale how the assumptions, threats, and organizational security policies are addressed by the objectives that are specified in the PP [6]. The following Table 11 reproduces the table in section 4.4 of [6].

Table 11: Security Objectives versus Assumptions, Threats or Policies

Assumption, Threat or OSP	Security Objective	Notes
A.Plat-Appl	OE.Plat-Appl	Phase 1
A.Resp-Appl	OE.Resp-Appl	Phase 1
P.Process-TOE	O.Identification	Phase 2 – 3
A.Process-Sec-IC	OE.Process-Sec-IC	Phases 4 - 6
T.Leak-Inherent	O.Leak-Inherent	
T.Phys-Probing	O.Phys-Probing	
T.Malfunction	O.Malfunction	
T.Phys-Manipulation	O.Phys-Manipulation	
T.Leak-Forced	O.Leak-Forced	
T.Abuse-Func	O.Abuse-Func	
T.RND	O.RND	

The rational provided in [6] is still applicable for Table 11. The following Table 12 provides the justification for the additional security objectives. They are in line with the security objectives of the PP [6] and supplement these according to the additional assumptions, threats, and organizational security policy.

Table 12: Additional Security Objectives versus Assumptions or Policies

Assumption/Policy	Security Objective	Note
A.Key-Function	OE.Plat-Appl OE.Resp-Appl	Phase 1
A.Check-Init	OE.Check-FabKey OE.Check-OriginalityKey	Phase 1 and 4 - 6 Phase 7
A.Secure-Values	OE.Secure-Values	Phase 5 to 7
A.Terminal-Support	OE.Terminal-Support	Phase 7
T.Data-Modification	O.DATA-ACCESS O.TYPE-CONSISTENCY	
T.Impersonate	O.AUTHENTICATION	This must be supported by OE.Secure-Values
T.Cloning	O.DATA-ACCESS O.AUTHENTICATION	This must be supported by OE.Secure-Values
P.Add-Components	O.HW-DES3 O.HW-AES O.MF-FW O.MEM-ACCESS O.SFR-ACCESS	
P.DESFire-Emulation	O.CONFIDENTIALITY O.TYPE-CONSISTENCY O.TRANSACTION	O.CONFIDENTIALITY requires OE.Terminal-Support

The justification related to the policies “Additional Specific Security Components (P.Add-Components and P.DESFire-Emulation)” is detailed below.

The policy P.Add-Components is related to the hardware platform and covers the additional objectives O.HW-DES3, O.HW-AES, O.MF-FW, O.MEM-ACCESS and O.SFR-ACCESS.

The justification related to these security objectives is as follows. Since these objectives require the TOE to implement exactly the same specific security functionality as required by P.Add-Components, the organizational security policy is covered by the objectives. The hardware platform is extended by the security functionality required by P.Add-Components. The extended security functionality provides also protection against the threats defined in the Protection Profile.

The policy P.DESFire-Emulation is related to the IC Dedicated Support Software and covers the additional objectives O.CONFIDENTIALITY, O.TYPE-CONSISTENCY, and O.TRANSACTION.

The justification related to these security objectives is as follows. Since these objectives require the TOE to implement exactly the same specific security functionality as required by P.DESFire-Emulation, the organizational security policy is covered by the objectives. The functionality of the hardware platform is extended by additional IC Dedicated Support Software that emulates the security functionality defined by the DESFire application. The extended security functionality provides also protection against the threats defined in the Protection Profile.

Therefore, 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-Components and P.DESFire-Emulation. These security objectives are also valid for the additional specific security functionality since they must avert the threats already defined in the Protection Profile also for the components added related to these policies.

The additional threats T.Data-Modification, T.Impersonate, T.Cloning are related to the DESFire application. They supplement the threats defined in the Protection Profile for the specific application context of the DESFire emulation. The threat T.Data-Modification is completely averted by the security objectives O.DATA-ACCESS and O.TYPE-CONSISTENCY provided by the TOE. The threat T.Impersonate is averted by the security objective O.AUTHENTICATION. This must be supported by OE.Secure-Values because the authentication is based on keys and the knowlegde of the keys must be limited to the authorized users. The threat T.Cloning is averted by O.DATA-ACCESS that prevents the disclosure of sensitive data from the TOE and O.AUTHENTICATION that limits the access to authorized user only. As already mentioned above, an appropriate key management according to OE.Secure-Values must be ensured. These additional threats are avert by the objectives introduced by P.DESFire-Emulation as summaries above.

The requirements for a multi-application platform necessitate the separation of users. Therefore it is volitional that most of the security functionality cannot be influenced or used in User Mode and that the Security IC Embedded Software and the IC Dedicated Software do not interfere. This is ensured by the additional objectives for the hardware platform since the separation of the memories and CPU modes is provided by the additional objectives introduced by P.Add-Components.

The justification related to the assumption A.Key-Function is as follows:

- Compared to [6] a clarification has been made for the security objective “Usage of Hardware Platform (OE.Plat-Appl)”: If required the Security IC Embedded Software shall use the cryptographic service of the TOE and its interface as specified. In addition, the Security IC Embedded Software (i) must implement operations on keys (if any) in such a manner that they do not disclose information about confidential data and (ii) must configure the memory management in a way that different applications are sufficiently separated. If the Security IC Embedded Software uses random numbers provided by the security service SS.HW_RNG these random numbers must be tested as appropriate for the intended purpose. This addition ensures that the assumption A.Key-Function is still covered by the objective OE.Plat-Appl although additional functions are being supported according to P.Add-Components.
- Compared to [6] 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 Security IC 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 implemented in the environment. In addition, the treatment of User Data comprises the implementation of a multi-application operating system that does not disclose security relevant User Data of one application to another one. These measures make sure that the assumption A.Key-Function is still covered by the security objective OE.Resp-Appl although additional functions are being supported according to P.Add-Components.

Since the security objectives for the environment OE.Plat Appl and OE.Resp Appl applies for all software running on the hardware platform the IC Dedicated Software as part of the TOE addresses these objectives to avert the threats defined in the Protection Profile and in this Security Target. Therefore the associated assumptions are covered.

The justification related to the assumption "Check of initialization data by the Security IC Embedded Software (A.Check-Init)" is split because the assumption is mapped to two security objectives for the environment. Both allow an appropriate identification of the TOE. The justification is as follows:

OE.Check-FabKey requires the Security IC Embedded Software developer to implement a function assumed in A.Check-Init to authenticate the TOE. This authentication is based on Fabkey data defined by the developer of the Security IC Embedded Software and stored in the Application-EEPROM. OE.Check-OriginalityKey requires the user of the DESFire EV1 Software to check the originality of the TOE as assumed in A.Check-Init. This check is based on data stored in the EEPROM and defined by NXP. Both security objectives for the environment are suitable to check the TOE as assumed in A.Check-Init. Based on the two security objectives for the environment the administrator of the DESFire EV1 Software and the user of the Security IC Embedded Software are independently able to identify the TOE.

The justification related to the assumption A.Secure_Values is as follows:

The management of the keys used for the authentication of roles for the DESFire application must be performed outside the TOE. These keys must be loaded in a personalization process and these keys must be protected by the environment. Since OE.Secure_Values requires from the Administrator, Application Manager or the Application User to use secure values for the configuration of the authentication and access control as assumed in A.Secure_Values, the assumption is covered by the objective.

The justification related to the assumption A.Terminal_Support is as follows:

The TOE can only check the integrity of data received from the terminal. For data transferred to the terminal the receiver must verify the integrity of the received data. This is assumed by the related assumption, therefore the assumption is covered.

5. Extended Components Definition

This Security Target does not define extended components.

Note that the PP “Security IC Platform Protection Profile” [6] defines extended security functional requirements in chapter 5, which are included in this Security Target.

6. Security Requirements

This chapter consists of the sections “Security Functional Requirements”, “Security Assurance Requirements” and “Security Requirements Rationale”.

6.1 Security Functional Requirements

The Security Functional Requirements (SFRs) of the TOE are presented in the following sections to support a better understanding of the combination of PP “Security IC Platform Protection Profile” [6] and Security Target.

6.1.1 SFRs of the Protection Profile

Table 13 below shows all SFRs, which are specified in the PP [6] (in the order of definition in the PP). Some of the SFRs are CC Part 2 extended and defined in the PP [6]. This is shown in the third column of the table.

Table 13: SFRs taken from the PP [6]

SFR	Title	Defined in
FRU_FLT.2	Limited fault tolerance	CC, Part 2
FPT_FLS.1	Failure with preservation of secure state	CC, Part 2
FMT_LIM.1	Limited capabilities	PP, Section 5.2
FMT_LIM.2	Limited availability	PP, Section 5.2
FAU_SAS.1	Audit storage	PP, Section 5.3
FPT_PHP.3	Resistance to physical attack	CC, Part 2
FDP_ITT.1	Basic internal transfer protection	CC, Part 2
FPT_ITT.1	Basic internal TSF data transfer protection	CC, Part 2
FDP_IFC.1	Subset information flow control	CC, Part 2
FCS_RNG.1	Random number generation	PP, Section 5.1

All operations except for the following assignments and selections are already performed in the PP [6].

For the SFR FAU_SAS.1 the PP [6] leaves the assignment operation open for the non-volatile memory type in which initialization data, pre-personalization data and/or other supplements for the Security IC Embedded Software are stored. This assignment operation is filled in by the following statement. Note that the assignment operations for the list of subjects and the list of audit information have already been filled in by the PP [6].

FAU_SAS.1 Audit storage

Hierarchical to: No other components.

Dependencies: No dependencies.

FAU_SAS.1.1 The TSF shall provide *the test process before TOE Delivery* [1] with the capability to store *the Initialization Data and/or Pre-*

[1] [assignment: *list of subjects*]

personalization Data and/or supplements of the Security IC Embedded Software [2] in the *EEPROM* [3].

For FCS_RNG.1.1 the PP [6] partially fills in the assignment for the security capabilities of the RNG by requiring a total failure test of the random source and adds an assignment operation for additional security capabilities of the RNG.

In addition, for FCS_RNG.1.2 the PP [6] partially fills in the assignment operation for the defined quality metric for the random numbers by replacing it by a selection and assignment operation.

For the above operations the original operations defined in chapter 5 of the PP [6] have been replaced by the open operations of the partially filled in operations in the statement of the security requirements in chapter 6 of [6] for better readability. Note that the selection operation for the RNG type has already been filled in by the PP [6].

FCS_RNG.1 Random number generation

Hierarchical to: No other components.

FCS_RNG.1.1 The TSF shall provide a *physical* [4] Random Number Generator that implements *total failure test of the random source and none* [5].

FCS_RNG.1.2 The TSF shall provide random numbers that meet *independent bits with Shannon entropy of 7.976 bits per octet*[6].

Dependencies: No dependencies.

Note: Application Note 20 in [6] requires that the Security Target specifies for the security capabilities in FCS_RNG.1.1 how the results of the total failure test of the random source are provided to the Security IC Embedded Software. The TOE features a hardware test which is called by the Security IC Embedded Software. The results of the internal test sequence are provided to the Security IC Embedded Software as a pass or fail criterion by means of a special function register.

The entropy of the random number is measured by the Shannon-Entropy as follows:

$E = -\sum_{i=0}^{255} p_i \cdot \log_2 p_i$, where p_i is the probability that the byte $(b_7, b_6, \ldots, b_0)$ is equal to i as binary number. Here term “bit” means measure of the Shannon-Entropy.

The value “7.976” is assigned due to the requirements of “AIS31”, [5].

By this, all assignment/selection operations are performed. This Security Target does not perform any other/further operations than stated in the PP [6].

[2] [assignment: *list of audit information*]

[3] [assignment: *type of persistent memory*]

[4] [selection: *physical, non-physical true, deterministic, hybrid*]

[5] [assignment: *list of additional security capabilities*]

[6] [selection: *independent bits with Shannon entropy of 7.976 bits per octet, Min-entropy of 7.95 bit per octet, [assignment: other comparable quality metric]*]

Considering the Application Note 12 of [6] in the following paragraphs the additional functions for cryptographic support and access control are defined. These SFRs are not required in the PP [6].

As required by the Application Note 14 of [6] the secure state is described in section 7.2.1 in the rationale for SF.OPC.

Regarding the Application Note 15 of [6] an additional generation of audit is not defined for “Limited fault tolerance” (FRU_FLT.2) and “Failure with preservation of secure state” (FPT_FLS.1).

As required by the Application Note 18 of [6] the automatic response of the TOE is described in section 7.2.1 in the rationale for SF.PHY.

6.1.2 Additional SFRs regarding cryptographic functionality

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

FCS_COP.1[HW_DES] Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform *encryption and decryption* [7] in accordance with a specified cryptographic algorithm *Triple Data Encryption Algorithm (TDEA)* [8] and cryptographic key sizes *of 112 or 168 bit* [9] that meet the following *list of standards* [10]:

FIPS PUB 46-3 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION DATA ENCRYPTION STANDARD (DES) Reaffirmed 1999 October 25, keying options 1 and 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.

Note: The cryptographic functionality FCS_COP.1[HW_DES] provided by the TOE achieves a security level of maximum 80 Bits, if keying option 2 is used.

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

FCS_COP.1[HW_AES] Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1 The TSF shall perform *encryption and decryption* [11] in accordance with a specified cryptographic algorithm

[7] [assignment: list of cryptographic operations]

[8] [assignment: cryptographic algorithm]

[9] [assignment: cryptographic key sizes]

[10] [assignment: list of standards]

[11] [assignment: list of cryptographic operations]

Advanced Encryption Standard (AES) algorithm [12] and cryptographic key sizes *of 128, 192 or 256 bit* [13] that meet the following *list of standards* [14]:

FIPS PUB 197 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, ADVANCED ENCRYPTION STANDARD (AES), National Institute of Standards and Technology, 2001 November 26.

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.

6.1.3 Additional SFRs regarding hardware access control

Hardware Access Control Policy

The hardware shall provide different CPU modes to the IC Dedicated Software and Security IC Embedded Software. The TOE shall separate IC Dedicated Software and Security IC Embedded Software from each other by both, partitioning of memory and different CPU modes. The management of access to code and data as well as the configuration of the hardware shall be performed each in a dedicated CPU mode. The hardware shall enforce a separation between different applications (i.e. parts of the Security IC Embedded Software) running on the TOE. An application shall not be able to access hardware components without explicitly granted permission.

The Security Function Policy (SFP) Hardware Access Control Policy uses the following definitions.

The subjects are

- The Security IC Embedded Software i.e. data in the memories of the TOE executed as instructions by the CPU
- The Test-ROM Software as IC Dedicated Test Software
- The Boot-ROM Software as part of the IC Dedicated Support Software
- The DESFire EV1 Software as part of the IC Dedicated Support Software

The objects are

- the memories consisting of
  - ROM, which is partitioned into Test-ROM and Application-ROM,
  - EEPROM, which is partitioned into two parts. For the ease of referencing the part reserved for the DESFire EV1 Software is called DESFire-EEPROM, the other part Application-EEPROM.
  - RAM, which is partitioned into two parts. For the ease of referencing the part reserved for the DESFire EV1 Software is called DESFire-RAM, the other part Application-RAM.
  - the code and data in the Memory Segments defined by the Memory Management Unit in Application-ROM, Application-EEPROM and Application-RAM. Note that this memory is a subset of the first three.

[12] [assignment: cryptographic algorithm]

[13] [assignment: cryptographic key sizes]

[14] [assignment: list of standards]

- the physical memory locations within the three memories that are used by the Memory Management Unit for the MMU Segment Table.
- the Special Function Registers consisting of
  - Special Function Registers to configure the MMU segmentation. This group contains the registers that define the pointer to the MMU Segment Table.
  - Special Function Registers related to system management, a number of Special Function Registers that are intended to be used for overall system management by the operating system.
  - Special Function Registers to configure the MIFARE firewall. These Special Function Registers allow to modify the MIFARE firewall regarding data exchange and Special Function Register access control.
  - Special Function Registers used by the DESFire EV1 Software. The DESFire EV1 Software uses a number of internal Special Function Registers.
  - Special Function Registers related to testing. These Special Function Registers are reserved for testing purposes.
  - Special Function Registers related to hardware components. These Special Function Registers are used to utilize hardware components like the coprocessors or the interrupt system.
  - Special Function Registers related to general CPU functionality. This group contains e.g. the accumulator, stack pointer and data pointers.

The memory operations are

- read data from the memory,
- write data into the memory and
- execute data in the memory.

The Special Function Register operations are

- read data from a Special Function Register and
- write data into a Special Function Register.

The security attributes are

- CPU mode: There are five CPU modes based on the configuration of the Special Function Register “Program Status Word High (PSWH)” and two internal bits defining whether the instruction is executed in Boot Mode, Test Mode, MIFARE Mode, System Mode or User Mode.
- The values of the Special Function Registers to configure the MMU segmentation and Special Function Registers related to system management. These groups contain the pointer to the MMU Segment Table and those relevant for the overall system management of the TOE, especially PSWH.
- MMU Segment Table: Configuration of the Memory Segments comprising access rights (read, write and execute), the virtual code memory base address of the first and last valid address, and the relocation offset to the physical memory location for each of the 64 possible Memory Segments. For every segment also the access rights to the Special Function Registers related to hardware components are defined.
- The values of the Special Function Registers FWCTRLL, FWCTRLH, MXBASL, MXBASH, MXSZL and MXSZH belonging to the group Special Function Registers to configure the MIFARE firewall that define the access rights to the Special Function Registers related to hardware components for code executed in MIFARE Mode and

the RAM area used for data exchange between IC Dedicated Support Software (DESFire EV1 Software) and Security IC Embedded Software.

In the following the term “code running” combined with a CPU mode (e.g. “code running in System Mode”) is used to name subjects.

Note: Use of a Memory Segment is disabled in case no access permissions are granted. It is not necessary to define all 64 possible Memory Segments, the Memory Management Unit is capable of managing an arbitrary number of segments up to the limit of 64.

The amount of the partitioned memory for EEPROM and RAM depends on the configuration of the TOE. DESFire EV Software supports three different size of the EEPROM (D2, D4, and D8), refer to section 1.4. 256 bytes of the EEPROM are always reserved for the manufacturer.

The TOE shall meet the requirements “Subset access control (FDP_ACC.1)” as specified below.

FDP_ACC.1[MEM] Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the *Hardware Access Control Policy* [15] on *all code running on the TOE, all memories and all memory operations* [16].

Dependencies: FDP_ACF.1 Security attribute based access control

Application Note: The Hardware Access Control Policy shall be enforced by implementing a Memory Management Unit, which maps virtual addresses to physical addresses. The CPU always uses virtual addresses, which are mapped to physical addresses by the Memory Management Unit. Prior to accessing the respective memory address, the Memory Management Unit checks if the access is allowed.

FDP_ACC.1[SFR] Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the *Hardware Access Control Policy* [17] on *all code running on the TOE, all Special Function Registers, and all Special Function Register operations* [18].

Dependencies: FDP_ACF.1 Security attribute based access control

Application Note: The Hardware Access Control Policy shall be enforced by implementing hardware access control to each Special Function Register. For every access the CPU mode is used to determine if the access shall be granted or denied. In addition, in User Mode and MIFARE Mode the access rights to the Special Function Registers related to hardware components are provided by the MMU Segment Table and the Special

[15] [assignment: access control SFP]

[16] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

[17] [assignment: access control SFP]

[18] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

Function Registers to configure the MIFARE firewall. A denied read access returns “0” instead of the actual value, a denied write access is in fact ignored. The read and/or write access to a Special Function Register may be not allowed depending on the function of the register or on the CPU mode to enforce the access control policy or ensure a secure operation.

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

FDP_ACF.1[MEM] **Security attribute based access control**

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the *Hardware Access Control Policy*[19] to objects based on the following: *all subjects and objects and the attributes CPU mode, the MMU Segment Table, the Special Function Registers to configure the MMU segmentation and the Special Function Registers related to system management* [20].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

Code executed in the Boot Mode

- *has read and execute access to all code/data in the Test-ROM,*
- *has read, write and execute access to all code/data in the DESFIRE-EEPROM*
- *has read and write access to all data in the DESFIRE-RAM*

Code executed in the Test Mode

- *has read and execute access to all code/data in the whole ROM,*
- *has read, write and execute access to all code/data in the whole EEPROM*
- *has read and write access to all data in the whole RAM*

Code executed in the MIFARE Mode

- *has read and execute access to all code/data in the Test-ROM,*
- *has read, write and execute access to all code/data in the DESFIRE-EEPROM*
- *has read and write access to all data in the DESFIRE-RAM*

Code executed in the System Mode

- *has read and execute access to all code/data in the Application-ROM,*

[19] [assignment: access control SFP]

[20] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]

- *has read, write and execute access to all code/data in the Application-EEPROM,*
- *has read and write access to all data in the Application-RAM,*

Code executed in the User Mode

- *has read and/or execute access to code/data in the Application-ROM controlled by the MMU Segment Table used by the Memory Management Unit,*
- *has read and/or write and/or execute access to code/data in the Application-EEPROM controlled by the MMU Segment Table used by the Memory Management Unit,*
- *has read and/or write access to data in the Application-RAM controlled by the MMU Segment Table used by the Memory Management Unit.* [21]

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *Code running in MIFARE Mode has read access to 64 bytes in the Application-ROM storing the “Access Condition Matrix”. Code running in MIFARE Mode has access to the Application-RAM defined by the Special Function Register MXBASL, MXBASH, MXSZL and MXSZH. Code running in Boot Mode or MIFARE Mode has read access to the Security Rows stored in the Application-EEPROM. The FameXE coprocessor has read access to the EEPROM and read/write access to the FameXE RAM.* [22]

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the *rules: no explicit rules* [23].

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

FDP_ACF.1[SFR] Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the *Hardware Access Control Policy* [24] to objects based on the following: *all subjects and objects and the attributes CPU mode, the MMU Segment Table and the Special Function Registers FWCTRLL and FWCTRLH* [25].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- *The code executed in Boot Mode is allowed to access all Special Function Register groups.*

[21] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

[22] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]

[23] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

[24] [assignment: access control SFP]

[25] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]

- *The code executed in Test Mode is allowed to access all Special Function Register groups.*
- *The code executed in MIFARE Mode is allowed to read Special Function Registers to configure the MIFARE firewall and to read/write Special Function Registers used by the DESFire EV1 Software. Access to Special Function Registers related to hardware components is based on the access rights determined by the Special Function Registers FWCTRLL and FWCTRLH.*
- *The code executed in System Mode is allowed to access Special Function Registers to configure the MMU segmentation, Special Function Registers related to system management, Special Function Registers to configure the MIFARE firewall and Special Function Registers related to hardware components.*
- *The code executed in the User Mode is allowed to access Special Function Registers related to hardware components based on the access rights defined in the respective Memory Segment in the MMU Segment Table from which the code is actually executed* [26]*.*

FDP_ACF.1.3	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: *In any CPU mode access to the Special Function Registers related to general CPU functionality is allowed. The Special Function Register PSWH belonging to group Special Function Registers related to system management is additionally readable in MIFARE Mode and User Mode. The Special Function Register CLKSEL of the group Special Function Registers related to hardware components can be read in the MIFARE Mode regardless of the MIFARE firewall settings given by FWCTRLL and FWCTRLH.* [27]
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the rules: *Access to Special Function Registers to configure the MMU segmentation is denied in all CPU modes except System Mode. The Special Function Registers RPT0, RPT1 and RPT2 of the group Special Function Registers related to system management are not readable. The Special Function Register RNR of the group Special Function Registers related to hardware components is read-only. The Special Function Registers AKEY and DKEY of the group Special Function Registers related to hardware components are not readable.* [28]
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization

[26] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

[27] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]

[28] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

Implications of the Hardware Access Control Policy

The Access Control Policy has some implications, that can be drawn from the policy and that are essential parts of the TOE security functionality.

- Code executed in Boot Mode or Test Mode is quite powerful. This code is used to configure and test the TOE.
- Code executed in the MIFARE Mode is separated from code executed in System Mode or User Mode. The separation is enforced by the partition of the memories provided by the Memory Management Unit. Only small memory areas are used for data exchange between the DESFire EV1 Software and the Security IC Embedded Software. Furthermore, the exchange area in RAM is fully controlled by code running in System Mode.
- Code executed in the System Mode can administrate the configuration of Memory Management Unit, because it has access to the respective Special Function Registers. Configuration means that the code can change the address of the MMU Segment Table and also modify the contents of it (as long as the table is located in write-able memory).
- Code executed in the User Mode cannot administrate the configuration of the Memory Management Unit, because it has no access to the Special Function Registers to configure the MMU segmentation. Therefore changing the pointer to the MMU Segment Table is not possible.
- It may be possible for User Mode code to modify the MMU Segment Table contents if the table itself is residing in a memory location that is part of a Memory Segment that the code has write access to.

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

FMT_MSA.3[MEM] Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the *Hardware Access Control Policy* [29] to provide *restrictive* [30] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow *no subject* [31] 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

Application Note: Restrictive means here that the reset values of the Special Function Register regarding the address of the MMU Segment Table are set to zero, which effectively disables any memory segment so that no User Mode code can be executed by the CPU. Furthermore, the memory partition cannot be configured at all.

The TOE does not provide objects or information that can be created, since it provides access to memory areas. The

[29] [assignment: access control SFP, information flow control SFP]

[30] [selection, choose one of: restrictive, permissive, [assignment: other property]]

[31] [assignment: the authorised identified roles]

definition of objects that are stored in the TOE’s memory is subject to the Security IC Embedded Software.

FMT_MSA.3[SFR] Static attribute initialization

Hierarchical to: No other components.

FMT_MSA.3.1 The TSF shall enforce the *Hardware Access Control Policy* [32] to provide *restrictive* [33] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow *no subject* [34] 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

Application Note: The TOE does not provide objects or information that can be created since no further security attributes can be derived (i.e. the set of Special Function Registers that contain security attributes is fixed). The definition of objects that are stored in the TOE’s memory is subject to the Security IC Embedded Software.

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

FMT_MSA.1[MEM] Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the *Hardware Access Control Policy* [35] to restrict the ability to *modify* [36] the security attributes *Special Function Registers to configure the MMU segmentation* [37] to *code executed in the System Mode* [38].

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

Application Note: The MMU Segment Table is not included in this requirement because it is located in the memory of the TOE and access to it is possible for every role that has access to the respective memory locations.

This component does not include any management functionality for the configuration of the memory partition. This is because the memory partition is fixed and cannot be changed after TOE delivery.

[32] [assignment: access control SFP, information flow control SFP]

[33] [selection, choose one of: restrictive, permissive, [assignment: other property]]

[34] [assignment: the authorised identified roles]

[35] [assignment: access control SFP(s), information flow control SFP(s)]

[36] [selection: change_default, query, modify, delete, [assignment: other operations]]

[37] [assignment: list of security attributes]

[38] [assignment: the authorised identified roles]

FMT_MSA.1[SFR] **Management of security attributes**

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the *Hardware Access Control Policy* [39] to restrict the ability to *modify* [40] the security attributes *defined in Special Function Registers* [41] to *code executed in a CPU mode which has write access to the respective Special Function Registers* [42].

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

FMT_SMF.1[HW] **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:

Change of the CPU mode by calling a system call vector (SVEC) or configuration vector (CVEC) address,

change of the CPU mode by invoking an exception or interrupt,

change of the CPU mode by finishing an exception/interrupt (with a RETI instruction),

change of the CPU mode with a special LCALL/ACALL/ECALL address,

change of the CPU mode by writing to the respective bits in the PSWH Special Function Register and

modification of the Special Function Registers containing security attributes, and

modification of the MMU Segment Table. [43]

Dependencies: No dependencies

Application Note: For the Hardware Access Control Policy the Security Functional Requirement FMT_MSA.1 is iterated. The dependency of FMT_MSA.1 to FMT_SMF.1 may imply a separation of the Specification of Management Functions. Iteration of FMT_SMF.1 is not needed because all management functions of the Hardware Access Control Policy rely on the same features implemented in the hardware.

6.1.4 Additional SFRs for the DESFire EV1 Software

The following policy and security functional requirements can only be provided by the TOE if the DESFire EV1 Software is called by the Security IC Embedded Software. The

[39] [assignment: access control SFP(s), information flow control SFP(s)]

[40] [selection: change_default, query, modify, delete, [assignment: other operations]]

[41] [assignment: list of security attributes]

[42] [assignment: the authorised identified roles]

[43] [assignment: list of management functions to be provided by the TSF]

subjects and objects described in the following policy are dedicated for the DESFire Emulation.

DESFire Access Control Policy

The Security Function Policy (SFP) DESFire Access Control Policy uses the following definitions:

The subjects are

- The **Administrator** i.e. the subject that owns or has access to the card master key.
- The **Application Manager** i.e. the subject that owns or has access to an application master key. Note that the TOE supports multiple applications and therefore multiple Application Managers, however for one application there is only one Application Manager.
- The **Application User** i.e. the subject that owns or has access to a key that allows to perform operations with application objects. Note that the TOE supports multiple Application Users within each application and the assigned rights to the Application Users can be different, which allows to have more or less powerful Application Users.
- Any other subject belongs to the role **Everybody**. This includes the card holder (i.e. end-user) and any other subject e.g. an attacker. These subjects do not possess any key and cannot perform operations that are restricted to the Administrator, Application Manager and Application User.
- The **Originality Key User** who can authenticate himself to prove the authenticity of the Security IC.
- The term **Nobody** will be used to explicitly indicate that no rights are granted to any subject.

The objects are

- The DESFire card level data itself.
- The DESFire EV1 Software can store a number of **Applications**.
- An application can store a number of **Data Files** of different types.
- One specific type of data file are **Values**.

Note that data files and values can be grouped in *standard files* and *backup files*, with values belonging to the group of backup files. When the term “file” is used without further information then both data files and values are meant.

The operations that can be performed with the objects are

- **read** a value or data from a data file,
- **write** data to a data file,
- **increase** a value (with a limit or unlimited),
- **decrease** a value,
- **create** an application, a value or a data file,
- **delete** an application, a value or a data file and
- **modify attribute** of the DESFire card level, an application, a value or a data file. Note that ‘freeze’ will be used as specific form of modification that prevents any further modify.

The security attributes are

- Attributes of the DESFire card level, applications, values and data files. There is a set of attributes for the DESFire card level, a set of attributes for every application and a set of attributes for every single file within an application. The term “**card attributes**” will be used for the set of attributes related to the DESFire card level, the term “**application attributes**” will be used for the set of application attributes and the term “**file attributes**” will be used for the attributes of values and data files.

Note that subjects are authorized by cryptographic keys. These keys are considered as authentication data and not as security attributes. The DESFire card level has a card master key. Every application has an application master key and a variable number of keys used for operations on data files or values (all these keys are called application keys).

The TOE shall meet the requirements “Subset access control (FDP_ACC.1)” as specified below.

FDP_ACC.1[DESFire] Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1 The TSF shall enforce the *DESFire Access Control Policy* [44] on *all subjects, objects, operations and attributes defined by the DESFire Access Control Policy* [45].

Dependencies: FDP_ACF.1 Security attribute based access control

Application Note: The DESFire Access Control Policy is related to the data maintained by the DESFire EV1 Software.

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

FDP_ACF.1[DESFire] Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1 The TSF shall enforce the *DESFire Access Control Policy* [46] to objects based on the following: *all subjects and objects and attributes* [47].

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

- *The Administrator can create and delete applications.*
- *The Application Manager of an application can create data file and values within this application, and delete data files and values within this application.*
- *An Application User can read or write a data file; read, increase or decrease a value based on the access control settings in the respective file attribute* [48].

[44] [assignment: access control SFP]

[45] [assignment: list of subjects, objects, and operations among subjects and objects covered by the SFP]

[46] [assignment: access control SFP]

[47] [assignment: list of subjects and objects controlled under the indicated SFP, and for each, the SFP-relevant security attributes, or named groups of SFP-relevant security attributes]

[48] [assignment: rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects]

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:

- *The Application Manager of an application can delete this application if this is allowed by a specific card attribute.*
- *Everybody can create applications if this is allowed by a specific card attribute.*
- *Everybody can create and delete data files or values of a specific application if this is allowed by a specific application attribute.*
- *Everybody can read or write a data file; read, increase or decrease a value if this is allowed by a specific file attribute* [49].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the *rules:*

- *Nobody can read or write a data file; read, increase or decrease a value if this is explicitly set for the respective operation on the respective data file or value* [50].

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

Implications of the DESFire Access Control Policy

The Access Control Policy has some implications, that can be drawn from the policy and that are essential parts of the TOE security functions.

- The TOE end-user does normally not belong to the group of authorized users (Administrator, Application Manager, Application User), but regarded as ‘Everybody’ by the TOE. This means that the TOE cannot determine if it is used by its intended end-user (in other words: it cannot determine if the current user is the owner of the Security IC).
- The Administrator can have the exclusive right to create and delete applications on the DESFire card level, however he can also grant this privilege to Everybody. Additionally, changing the card attributes is reserved for the Administrator. Application keys, at delivery time should be personalized to a preliminary, temporary key only known to the Administrator and the Application Manager.
- At application personalization time, the Application Manager uses the preliminary application key in order to personalize the application keys, whereas all keys, except the application master key, can be personalized to a preliminary, temporary key only known to the Application Manager and the Application User. Furthermore, the Application Manager has the right to create files within his application scope.

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

FMT_MSA.3[DESFire] Static attribute initialization

Hierarchical to: No other components.

[49] [assignment: rules, based on security attributes, that explicitly authorise access of subjects to objects]

[50] [assignment: rules, based on security attributes, that explicitly deny access of subjects to objects]

FMT_MSA.3.1 The TSF shall enforce the *DESFire Access Control Policy* [51] to provide *permissive* [52] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow *no subject* [53] 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

Application Note: The only initial attributes are the card attributes. All other attributes have to be defined at the same time the respective object is created.

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

FMT_MSA.1[DESFire] Management of security attributes

Hierarchical to: No other components.

FMT_MSA.1.1 The TSF shall enforce the *DESFire Access Control Policy* [54] to restrict the ability to *modify or freeze* [55] the security attributes *card attributes, application attributes and file attributes* [56] to *the Administrator, Application Manager and Application User, or Everybody* [57] based on the refinement below.

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

Refinement: The detailed management abilities are:

- The Administrator can modify the card attributes. The card attributes contain a flag that when set will prevent any further change of the card attributes, thereby allowing to freeze the card attributes.
- The Application Manager can modify the application attributes. The application attributes contain a flag that when set will prevent any further change of the application attributes, thereby allowing to freeze the application attributes.
- The Application Manager can decide to restrict the ability to modify the file attributes to the Application Manager, an Application User, Everybody or to Nobody. The restriction to Nobody is equivalent to freezing the file attributes.

[51] [assignment: access control SFP, information flow control SFP]

[52] [selection, choose one of: restrictive, permissive, [assignment: other property]]

[53] [assignment: the authorised identified roles]

[54] [assignment: access control SFP, information flow control SFP]

[55] [selection: change_default, query, modify, delete, [assignment: other operations]]

[56] [assignment: list of security attributes]

[57] [assignment: the authorised identified roles]

- As an implication of the last rule, any subject that receives the modify abilities from the Application Manger gets these abilities transferred.
- The implication given in the previous rule includes the possibility for an Application User to modify the file attributes if the Application Manager decides to transfer this ability. If there is no such explicit transfer, an Application User does not have the ability to modify the file attributes.

The TOE shall meet the requirement “Specification of Management Functions (FMT_SMF.1)” as specified below.

FMT_SMF.1[DESFire] **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:

Authenticate a user,

Invalidating the current authentication state based on the functions: Selecting an application or the DESFire card level, Changing a key, Occurrence of any error during the execution of a command, Reset;

Changing a security attribute,

Creating or deleting an application, a value or a data file. [58]

Dependencies: No dependencies

The TOE shall meet the requirements “Security roles (FMT_SMR.1)” as specified below.

FMT_SMR.1[DESFire] **Security roles**

Hierarchical to: No other components.

FMT_SMR.1.1 The TSF shall maintain the roles *Administrator, Application Manager, Application User, Everybody and Originality Key User* [59].

FMT_SMR.1.2 The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

Note: The SFR FMT_SMR.1 includes the identifier “[DESFire]” since it is a dependency of other SFR and thereby it is shown with dependencies is satisfied. Nobody is not considered as role and therefore not included in the list above.

The TOE shall meet the requirement “Import of user data with security attributes (FDP_ITC.2)” as specified below.

FDP_ITC.2 **Import of user data with security attributes**

Hierarchical to: No other components.

FDP_ITC.2.1 The TSF shall enforce the *DESFire Access Control Policy* [60] when importing user data, controlled under the SFP, from outside of the TOE.

[58] [assignment: list of security management functions to be provided by the TSF]

[59] [assignment: the authorised identified roles]

FDP_ITC.2.2 The TSF shall use the security attributes associated with the imported user data.

FDP_ITC.2.3 The TSF shall ensure that the protocol used provides for the unambiguous association between the security attributes and the user data received.

FDP_ITC.2.4 The TSF shall ensure that interpretation of the security attributes of the imported user data is as intended by the source of the user data.

FDP_ITC.2.5 The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TOE: *no additional rules* [61].

Dependencies: [FDP_ACC.1 Subset access control, or
FDP_IFC.1Subset information flow control]
[FTP_ITC.1 Inter-TSF trusted channel, or
FTP_TRP.1 Trusted path]
FPT_TDC.1 Inter-TSF basic TSF data consistency

The TOE shall meet the requirement “Inter-TSF basic TSF data consistency (FPT_TDC.1)” as specified below.

FPT_TDC.1 Inter-TSF basic TSF data consistency

Hierarchical to: No other components.

FPT_TDC.1.1 The TSF shall provide the capability to consistently interpret *data files and values*[62] when shared between the TSF and another trusted IT product.

FPT_TDC.1.2 The TSF shall use *the rule: data files or values can only be modified by their dedicated type-specific operations honoring the type-specific boundaries*[63] when interpreting the TSF data from another trusted IT product.

Dependencies: No dependencies.

Note: The TOE does not interpret the *contents* of the data, e.g. it cannot determine if data stored in a specific data file is an identification number that adheres to a specific format. Instead the TOE distinguishes different types of files and ensures that type-specific boundaries cannot be violated, e.g. values do not overflow, single records are limited by their size and cyclic records are handled correctly.

The TOE shall meet the requirement “User identification before any action (FIA_UID.2)” as specified below.

FIA_UID.2 User identification before any action

Hierarchical to: FIA_UID.1

[60] [assignment: access control SFP and/or information flow control SFP]

[61] [assignment: additional importation control rules]

[62] [assignment: list of TSF data types]

[63] [assignment: list of interpretation rules to be applied by the TSF]

FIA_UID.2.1	The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	No dependencies.
Application Note:	Identification of a user is performed upon an authentication request based on the currently selected context and the key number. For example, if an authentication request for key number 0 is issued after selecting a specific application, the user is identified as the Application Manager of the respective application. Before any authentication request is issued the user is identified as 'Everybody'.

The TOE shall meet the requirement "User authentication before any action (FIA_UAU.2)" as specified below.

FIA_UAU.2 User authentication before any action

Hierarchical to:	FIA_UAU.1
FIA_UAU.2.1	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	FIA_UID.1 Timing of identification

The TOE shall meet the requirement "Multiple authentication mechanisms (FIA_UAU.5)" as specified below.

FIA_UAU.5 Multiple authentication mechanisms

Hierarchical to: No other components.

FIA_UAU.5.1 The TSF shall provide *'none' and cryptographic authentication* [64] to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the *following rules:*

- *The 'none' authentication is performed with anyone who communicates with the TOE without issuing an explicit authentication request. The 'none' authentication implicitly and solely authorizes the 'Everybody' subject.*
- *The cryptographic authentication is used to authorize the Administrator, Application Manager, Application User and Originality Key User.* [65].

Dependencies: No dependencies.

The TOE shall meet the requirement "Management of TSF data (FMT_MTD.1)" as specified below.

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1 The TSF shall restrict the ability to *change_default, modify or freeze* [66] the *card master key, application master keys and*

[64] [assignment: list of multiple authentication mechanisms]

[65] [assignment: rules describing how the multiple authentication mechanisms provide authentication]

application keys [67] to *the Administrator, Application Manager and Application User* [68].

Refinement: The detailed management abilities are:

- The originality key cannot be changed by any role.
- The Administrator can modify the card master key. The card attributes contains a flag that when set will prevent any further change of the card master key, thereby allowing to freeze the card master key.
- The Administrator can change the default key that is used as the application master key and for the application keys when an application is created.
- The Application Manager of an application can modify the application master key of the application assigned to him. The application attributes contain a flag that when set will prevent any further change of the application master key, thereby allowing to freeze the application master key.
- The Application Manager can decide to restrict the ability to modify the application keys to the Application Manager, the Application Users or to Nobody. The restriction to Nobody is equivalent to freezing the application keys.
- The Application Users can either change their own keys or one Application User can be defined that can change all keys of the Application Users within an application.
- As an implication of the last rule, any subject that receives the modify abilities from the Application Manger gets these abilities transferred.
- The Originality Key User can only access the originality key to verify the originality of the Security IC.

Dependencies: FMT_SMR.1 Security roles
FMT_SMF.1 Specification of Management Functions

The TOE shall meet the requirement “Trusted path (FTP_TRP.1)” as specified below.

FTP_TRP.1 Trusted path

Hierarchical to: No other components.

FTP_TRP.1.1 The TSF shall provide a communication path between itself and *remote* [69] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from *modification or disclosure* [70].

[66] [selection: change_default, query, modify, delete, clear, [assignment: other operations]]

[67] [assignment: list of TSF data]

[68] [assignment: the authorised identified roles]

[69] [selection: remote, local]

[70] [selection: modification, disclosure, [assignment: other types of integrity or confidentiality violation]]

FTP_TRP.1.2	The TSF shall permit *remote users* [71] to initiate communication via the trusted path.
FTP_TRP.1.3	The TSF shall require the use of the trusted path for *authentication requests with DES or AES, confidentiality and/or data integrity verification for data transfers protected with AES and based on a setting in the file attributes* [72].
Dependencies:	No dependencies.

The TOE shall meet the requirement “Cryptographic key destruction (FCS_CKM.4)” as specified below.

FCS_CKM.4	**Cryptographic key destruction**
Hierarchical to:	No other components.
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *overwriting of memory* [73] that meets the following: *none* [74].
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] FMT_MSA.2 Secure security attributes

The TOE shall meet the requirement “Basic rollback (FDP_ROL.1)” as specified below.

FDP_ROL.1 **Basic rollback**

Hierarchical to: No other components.

FDP_ROL.1.1	The TSF shall enforce *DESFire Access Control Policy* [75] to permit the rollback of the *operations that modify the value or data file objects* [76] on the *backup files* [77].
FDP_ROL.1.2	The TSF shall permit operations to be rolled back within the *scope of the current transaction, which is defined by the following limitative events: chip reset, (re-)authentication (either successful or not), select command, explicit commit, explicit abort, command failure* [78].
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]

The TOE shall meet the requirement “Replay detection (FPT_RPL.1)” as specified below.

FPT_RPL.1	**Replay detection**
Hierarchical to:	No other components.
FPT_RPL.1.1	The TSF shall detect replay for the following entities: *authentication requests with DES or AES, confidentiality*

[71] [selection: the TSF, local users, remote users]

[72] [selection: initial user authentication,[assignment: other services for which trusted path is required]]

[73] [assignment: cryptographic key destruction method]

[74] [assignment: list of standards]

[75] [assignment: access control SFP(s) and/or information flow control SFP(s)]

[76] [assignment: list of operations]

[77] [assignment: information and/or list of objects]

[78] [assignment: boundary limit to which rollback may be performed]

	and/or data integrity verification for data transfers protected with AES and based on a setting in the file attributes [79].
FPT_RPL.1.2	The TSF shall perform *rejection of the request* [80] when replay is detected.
Dependencies:	No dependencies.

6.2 Security Assurance Requirements

Table 14 below lists all security assurance components that are valid for this Security Target. With one exception these security assurance components are required by EAL4 (see section 2.2) or by the PP “Security IC Platform Protection Profile” [6].

The exception is the component ASE_TSS.2 which is chosen as an augmentation in this Security Target to give architectural information on the security functionality of the TOE.

Considering Application Note 21 of [6] the column “Required by” shows the differences in the requirements of security assurance components between the PP [6] and the Security Target. The entry “EAL4 / PP” denotes, that an SAR is required by both EAL4 and the requirement of the PP [6], “PP” identifies this component as a requirement of the PP which is beyond EAL4. The augmentation ASE_TSS.2 chosen in this Security Target is denoted by "ST". The refinements of the PP [6] are considered in this Security Target.

Table 14: Security Assurance Requirements

SAR	Title	Required by
ADV_ARC.1	Security architecture description	EAL4 / PP
ADV_FSP.4	Complete functional specification	EAL4 / PP
ADV_IMP.1	Implementation representation of the TSF	EAL4 / PP
ADV_TDS.3	Basic modular design	EAL4 / PP
AGD_OPE.1	Operational user guidance	EAL4 / PP
AGD_PRE.1	Preparative procedures	EAL4 / PP
ALC_CMC.4	Production support, acceptance procedures and automation	EAL4 / PP
ALC_CMS.4	Problem tracking CM coverage	EAL4 / PP
ALC_DEL.1	Delivery procedures	EAL4 / PP
ALC_DVS.2	Sufficiency of security measures	PP
ALC_LCD.1	Developer defined life-cycle model	EAL4 / PP
ALC_TAT.1	Well-defined development tools	EAL4 / PP
ASE_CCL.1	Conformance claims	EAL4 / PP
ASE_ECD.1	Extended components definition	EAL4 / PP

[79] [assignment: list of identified entities]

[80] [assignment: list of specific actions]

SAR	Title	Required by
ASE_INT.1	ST introduction	EAL4 / PP
ASE_OBJ.2	Security objectives	EAL4 / PP
ASE_REQ.2	Derived security requirements	EAL4 / PP
ASE_SPD.1	Security problem definition	EAL4 / PP
ASE_TSS.2	TOE summary specification with architectural design summary	ST
ATE_COV.2	Analysis of coverage	EAL4 / PP
ATE_DPT.2	Testing: security enforcing modules	EAL4 / PP
ATE_FUN.1	Functional testing	EAL4 / PP
ATE_IND.2	Independent testing - sample	EAL4 / PP
AVA_VAN.5	Advanced methodical vulnerability analysis	PP

6.3 Security Requirements Rationale

6.3.1 Rationale for the security functional requirements

Section 6.3.1 in [6] provides a rationale for the mapping between security functional requirements and security objectives defined in the PP [6]. The mapping is reproduced in the following table.

Table 15: Security Requirements versus Security Objectives

Objective	TOE Security Functional Requirements
O.Leak-Inherent	FDP_ITT.1 “Basic internal transfer protection” FPT_ITT.1 “Basic internal TSF data transfer protection” FDP_IFC.1 “Subset information flow control”
O.Phys-Probing	FPT_PHP.3 “Resistance to physical attack”
O.Malfunction	FRU_FLT.2 “Limited fault tolerance FPT_FLS.1 “Failure with preservation of secure state”
O.Phys-Manipulation	FPT_PHP.3 “Resistance to physical attack”
O.Leak-Forced	All requirements listed for O.Leak-Inherent FDP_ITT.1, FPT_ITT.1, FDP_IFC.1 plus those listed for O.Malfunction and O.Phys-Manipulation FRU_FLT.2, FPT_FLS.1, FPT_PHP.3
O.Abuse-Func	FMT_LIM.1 “Limited capabilities” FMT_LIM.2 “Limited availability” plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3, FRU_FLT.2,

Objective	TOE Security Functional Requirements
	FPT_FLS.1
O.Identification	FAU_SAS.1 “Audit storage”
O.RND	FCS_RNG.1 “Quality metric for random numbers” plus those for O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation, O.Leak-Forced FDP_ITT.1, FPT_ITT.1, FDP_IFC.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1
OE.Plat-Appl	not applicable
OE.Resp-Appl	not applicable

The Security Target additionally defines SFRs for the hardware platform as well as for the DESFire EV1 Software of the TOE. These SFR are mapped to the additional objectives in the following table. The table gives an overview, how the requirements are combined to meet the security objectives.

Table 16: Mapping of security objectives and requirements

Objective	TOE Security Functional Requirement	Note
O.HW-DES3	FCS_COP.1[HW_DES]	Policy P.Add-Components
O.HW-AES	FCS_COP.1[HW_AES]	Policy P.Add-Components
O.MF-FW	FDP_ACC.1[MEM] FDP_ACF.1[MEM] FMT_MSA.3[MEM]	Policy P.Add-Components
O.MEM-ACCESS	FDP_ACC.1[MEM] FDP_ACF.1[MEM] FMT_MSA.3[MEM] FMT_MSA.1[MEM] FMT_MSA.1[SFR] FMT_SMF.1[HW]	Policy P.Add-Components
O.SFR-ACCESS	FDP_ACC.1[SFR] FDP_ACF.1[SFR] FMT_MSA.3[SFR] FMT_MSA.1[SFR] FMT_SMF.1[HW]	Policy P.Add-Components
O.DATA-ACCESS	FMT_SMR.1[DESFire] FDP_ACC.1[DESFire] FDP_ACF.1[DESFire] FMT_MSA.3[DESFire] FMT_MSA.1[DESFire] FMT_SMF.1[DESFire] FDP_ITC.2 FCS_CKM.4 FMT_MTD.1	Policy P.DESFire-Emulation
O.AUTHENTICATION	FCS_COP.1[HW_DES]	Policy P.DESFire-Emulation

Objective	TOE Security Functional Requirement	Note
	FCS_COP.1[HW_AES] FIA_UID.2 FIA_UAU.2 FIA_UAU.5 FTP_TRP.1 FPT_RPL.1	
O.CONFIDENTIALITY	FCS_COP.1[HW_AES] FTP_TRP.1 FPT_RPL.1	Policy P.DESFire-Emulation
O.TYPE-CONSISTENCY	FPT_TDC.1	Policy P.DESFire-Emulation
O.TRANSACTION	FDP_ROL.1	Policy P.DESFire-Emulation
OE.Check-FabKey	not applicable	
OE.Check-OriginalityKey	not applicable	
OE.Secure-Values	not applicable	
OE.Terminal-Support	not applicable	

The additional security objectives are related to the additional security functionality provided by the hardware platform derived from P.Add-Components and the security functionality provided by the IC Dedicated Support Software derived from P.DESFire-Emulation. In the following the rationale for the components derived from policies P.Add-Components and P.DESFire-Emulation are given:

Rationale for the requirements introduced by **P.Add-Components**:

The justification related to the security objective “Triple DES Functionality” **(O.HW-DES3)** is as follows:

O.HW-DES3 requires the TOE to support Triple DES encryption and decryption. Exactly this is the requirement of FCS_COP.1[HW_DES]. Therefore FCS_COP.1[HW_DES] is suitable to meet O.HW-DES3.

The justification related to the security objective “AES Functionality” **(O.HW-AES)** is as follows:

O.HW-AES requires the TOE to support AES encryption and decryption. Exactly this is the requirement of FCS_COP.1[HW_AES]. Therefore FCS_COP.1[HW_AES] is suitable to meet O.HW-AES.

The justification related to the security objective “MIFARE Firewall” (**O.MF-FW**) is as follows:

The security functional requirement “Subset access control (FDP_ACC.1[MEM])” with the related Security Function Policy (SFP) “Hardware Access Control Policy” exactly require to implement a memory partition as demanded by O.MF-FW. Therefore, FDP_ACC.1[MEM] with its SFP is suitable to meet the security objective.

The security functional requirement “Security attribute based access control (FDP_ACF.1[MEM])” with the related Security Function Policy (SFP) “Access Control Policy” defines the rules to implement the partition as demanded by O.MF-FW. Therefore, FDP_ACF.1[MEM] with its SFP is suitable to meet the security objective.

The security functional requirement “Static attribute initialization (FMT_MSA.3[MEM])” requires that the TOE provide default values for the security attributes used by the Memory Management Unit to enforce the memory partition. These default values are configured by the reset procedure and the Boot-ROM Software. Restrictive with respect to memory partition means that the partition cannot be changed at all. Therefore this requirement (as dependency from FDP_ACF.1) is suitable to meet the security objective.

The security functional requirement “Management of security attributes (FMT_MSA.1)” requires that the ability to update the security attributes is restricted to privileged subject(s). No management ability is specified in the two iterations of FMT_MSA.1 for the objective O.MF-FW because the memory partitioning shall not be changed. Therefore the memory partition is fixed and cannot be changed any subject, as required by O.MF-FW.

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[MEM])” with the related Security Function Policy (SFP) “Hardware Access Control Policy” exactly require to implement an area based memory access control as demanded by O.MEM-ACCESS. Therefore, FDP_ACC.1[MEM] with its SFP is suitable to meet the security objective.

The security functional requirement “Security attribute based access control (FDP_ACF.1[MEM])” with the related Security Function Policy (SFP) “Access Control Policy” defines the rules to implement the area based memory access control as demanded by O.MEM-ACCESS. Therefore, FDP_ACF.1[MEM] with its SFP is suitable to meet the security objective.

The security functional requirement “Static attribute initialization (FMT_MSA.3[MEM])” requires that the TOE provide default values for the security attributes used by the Memory Management Units. Restrictive with respect to the memory segmentation means that the initial setting is very restrictive since it effectively disables any memory segment. Since the TOE is a hardware platform these default values are generated by the reset procedure for the related Special Function Register. Therefore this requirement (as dependency from FDP_ACF.1) is suitable to meet the security objective.

The security functional requirement “Management of security attributes (FMT_MSA.1)” requires that the ability to update the security attributes is restricted to privileged subject(s). These management functions ensure that the required access control can be realized using the functions provided by the TOE. The iteration of FMT_MSA.1 into FMT_MSA.1[MEM] and FMT_MSA.1[SFR] is needed because the different types of objects have different security attributes. The security attributes of the Memory Management Unit can be changed by the Security IC Embedded Software. Since the pointer to the MMU Segment Table can only be changed in System Mode and this protection is implemented by access control to the respective Special Function Registers, both iterations are needed for O.MEM-ACCESS.

Finally, the security functional requirement “Specification of Management Functions (FMT_SMF.1)” is used for the specification of the management functions to be provided by the TOE as demanded by O.MEM-ACCESS. Therefore, FMT_SMF.1[HW] is suitable to meet the security objective.

The justification related to the security objective “Special Function Register Access Control (**O.SFR-ACCESS**)” is as follows:

The security functional requirement “Subset access control (FDP_ACC.1[SFR])” with the related Security Function Policy (SFP) “Hardware Access Control Policy” require to implement access control for Special Function Register as demanded by O.SFR-

ACCESS. Therefore, FDP_ACC.1[SFR] with its SFP is suitable to meet the security objective.

The access to Special Function Register is related to the CPU mode. The Special Function Register used to configure the Memory Management Unit can only be accessed in the System Mode. The Special Function Register required to use hardware components like e.g. the coprocessors or the Random Number Generator can be accessed in the System Mode as specified by the Security Function Policy (SFP) “Hardware Access Control Policy”. In the User Mode only Special Function Register required to run the CPU are accessible by default. In addition, specific Special Function Registers related to hardware components can be made accessible for the User Mode if the Memory Management Unit is configured to allow this.

The security functional requirement “Security attribute based access control (FDP_ACF.1[SFR])” with the related Security Function Policy “Access Control Policy” exactly requires certain security attributes to implement the access control to Special Function Register as demanded by O.SFR-ACCESS. Therefore, FDP_ACF.1[SFR] with its SFP is suitable to meet the security objective.

The security functional requirement “Static attribute initialization (FMT_MSA.3[SFR])” requires that the TOE provides default values for the Special Function Register (values as well as access control). The default values are needed to ensure a defined setup for the operation of the TOE. Therefore this requirement (as dependency from FDP_ACF.1) is suitable to meet the security objective.

The security functional requirement “Management of security attributes (FMT_MSA.1[SFR])” is realized in a way that – besides the definition of access rights to Special Function Registers related to hardware components in User Mode and MIFARE Mode - no management of the security attributes is possible because the attributes are implemented in the hardware and cannot be changed.

Finally, the security functional requirement “Specification of Management Functions (FMT_SMF.1)” is used for the specification of the management functions to be provided by the TOE as demanded by O.SFR-ACCESS. Therefore, FMT_SMF.1[HW] is suitable to meet the security objective.

Note that the iteration of FDP_ACF.1 and FDP_ACC.1 with the respective dependencies is needed to separate the different types of objects because they have different security attributes. Since the management functions of the SFP Hardware Access Control Policy regarding O.MEM-ACCESS and O.SFR-ACCESS cannot be split the security functional requirement is assigned to both iterations.

Rationale for the requirements introduced by **P.DESFire-Emulation**:

The justification related to the security objective “Access control to data controlled by the DESFire EV1 Software (**O.DATA-ACCESS**)” is as follows:

The SFR FMT_SMR.1 defines the roles of the Access Control Policy. The SFR FDP_ACC.1[DESFIRE] and FDP_ACF.1[DESFIRE] define the rules. The initialization and change of the attributes for the access control is covered by FMT_MSA.3[DESFIRE]. FMT_MTD.1 provides the rules for the management of the authentication data. The management functions are defined by FMT_SMF.1[DESFIRE]. The TOE stores data on behalf of the authorized subjects importing user data with security attributes as defined by FDP_ITC.2. Since cryptographic keys are used for authentication (refer to O.AUTHENTICATION), these keys have to be removed if the access permission based on such a key are related to a specific application is not longer needed (i.e. an application is deleted). This is required by FCS_CKM.4. These SFR provide an access control mechanism as required by the objective O.DATA-ACCESS.

The justification related to the security objective “User Authentication for the DESFire EV1 Software (**O.AUTHENTICATION**)” is as follows:

The identification and authentication is required by SFR FIA_UID.2 and FIA_UAU.2. Further FIA_UAU.5 requires multiple authentication mechanisms. Together these SFR define that users must be identified and authenticated before any action. The ‘none’ authentication of FIA_UAU.5 also ensures that a specific subject is identified and authenticated before an explicit authentication request is sent to the TOE. FTP_TRP.1 requires a trusted communication path between the TOE and remote users, FTP_TRP.1.3 especially requires “authentication requests”. FPT_RPL.1 requires a replay detection for these authentication requests. Therefore these SFR implement the functionality as required by O.AUTHENTICATION. This is supported by the cryptographic coprocessors provided by the hardware platform, refer to FCS_COP.1[HW_DES] and FCS_COP.1[HW_AES].

The justification related to the security objective “Protection of the communication (**O.CONFIDENTIALITY**)” is as follows:

FTP_TRP.1 requires a trusted communication path between the TOE and remote users, FTP_TRP.1.3 especially requires “authentication requests”. The SFR FPT_RPL.1 requires replay detection for integrity control and confidentiality of data as well as authentication requests. Therefore these SFR implement the functionality as required by O.CONFIDENTIALITY. This is supported by the cryptographic coprocessors provided by the hardware platform, refer to FCS_COP.1[HW_AES].

The justification related to the security objective “Consistent handling of data types (**O.TYPE-CONSISTENCY**)” is as follows:

The SFR FPT_TDC.1 requires the TOE to consistently interpret data files and values. The TOE will honor the respective file formats and boundaries (i.e. upper and lower limits, size limitations). This meets the objective O.TYPE-CONSISTENCY.

The justification related to the security objective “Transaction control (**O.TRANSACTION**)” is as follows:

The SFR FDP_ROL.1 requires the possibility to rollback a set of modifying operations on backup files in total. The set of operations is defined by the scope of the transaction, which is itself limited by boundary events. This fulfils the objective O.TRANSACTION.

The justification related to the security objective for the operational environment given below Table 12 is considered to be complete also for this aspect.

6.3.2 Dependencies of security functional requirements

The dependencies listed in the PP [6] are independent of the additional dependencies listed in the table below. The dependencies identified in the PP [6] are fulfilled based on the Security Functional Requirements included in the PP [6]. One dependency of the Security Functional Requirements defined in the PP is open however based on the discussion in the PP the dependency is considered to be satisfied and the discussion is applicable for this Security Target.

This Security Target includes additional Security Functional Requirements as defined in Part 2 of the Common Criteria. The following discussion demonstrates how the dependencies for these requirements specified in sections 6.1.2 and 6.1.3 are satisfied.

The dependencies defined in the Common Criteria are listed in the table below. The table includes only those Security Functional Requirements used in the Security Target, for which dependencies are defined.

Table 17: Dependencies of security functional requirements

Security Functional Requirement	Dependencies	Fulfilled by security requirements in this ST
FCS_COP.1[HW_DES]	FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1 FCS_CKM.4	No, see discussion below
FCS_COP.1[HW_AES]	FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1 FCS_CKM.4	No, see discussion below
FDP_ACC.1[MEM]	FDP_ACF.1	Yes, by FDP_ACF.1[MEM]
FDP_ACC.1[SFR]	FDP_ACF.1	Yes, by FDP_ACF.1[SFR]
FDP_ACF.1[MEM]	FDP_ACC.1 FMT_MSA.3	Yes, by FDP_ACC.1[MEM] Yes
FDP_ACF.1[SFR]	FDP_ACC.1 FMT_MSA.3	Yes, by FDP_ACC.1[SFR] Yes
FMT_MSA.3[MEM]	FMT_MSA.1 FMT_SMR.1	Yes, by FMT_MSA.1[MEM] No, see discussion below
FMT_MSA.3[SFR]	FMT_MSA.1 FMT_SMR.1	Yes, by FMT_MSA.1[SFR] No, see discussion below
FMT_MSA.1[MEM]	FDP_ACC.1 or FDP_IFC.1 FMT_SMR.1 FMT_SMF.1	Yes, by FDP_ACC.1[MEM] No, see discussion below Yes
FMT_MSA.1[SFR]	FDP_ACC.1 or FDP_IFC.1 FMT_SMR.1 FMT_SMF.1	Yes, by FDP_ACC.1[SFR] No, see discussion below Yes
FDP_ACC.1[DESFIRE]	FDP_ACF.1	Yes, by FDP_ACF.1[DESFIRE]
FDP_ACF.1[DESFIRE]	FDP_ACC.1 FMT_MSA.3	Yes, by FDP_ACC.1[DESFIRE] Yes
FMT_MSA.3[DESFIRE]	FMT_MSA.1 FMT_SMR.1	Yes, by FMT_MSA.1[DESFIRE] Yes
FMT_MSA.1[DESFIRE]	FDP_ACC.1 or FDP_IFC.1 FMT_SMR.1 FMT_SMF.1	Yes, by FDP_ACC.1[DESFIRE] Yes Yes
FMT_SMR.1[DESFIRE]	FIA_UID.1	Yes (by FIA_UID.2)

Security Functional Requirement	Dependencies	Fulfilled by security requirements in this ST
FDP_ITC.2	FDP_ACC.1 or FDP_IFC.1 FTP_ITC.1 or FTP_TRP.1 FPT_TDC.1	Yes, by FDP_ACC.1[DESFIRE] Yes, by FTP_TRP.1 Yes
FIA_UAU.2	FIA_UID.1	Yes (by FIA_UID.2)
FMT_MTD.1	FMT_SMR.1 FMT_SMF.1	Yes Yes
FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 FMT_MSA.2	Yes (by FDP_ITC.2) No, see discussion below
FDP_ROL.1	FDP_ACC.1 or FDP_IFC.1	Yes, by FDP_ACC.1[DESFIRE]

The dependent requirements of FCS_COP.1[HW_DES] and FCS_COP.1[HW_AES] completely address the appropriate management of cryptographic keys used by the specified cryptographic function and the management of access control rights for these keys. The functional requirements [FDP_ITC.1, or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4 are not included for the hardware platform since the hardware platform only provides a pure engine for encryption and decryption without additional features for the handling of cryptographic keys. Access control can be provided by the TOE as specified in the Hardware Access Control Policy. All requirements concerning these management functions shall be fulfilled by the environment (Security IC Embedded Software). The user management and the access control to the keys depend on the application context, therefore the Security IC Embedded Software must fulfill these requirements related to the needs of the realized application.

The dependency FMT_SMR.1 introduced by the components FMT_MSA.1 and FMT_MSA.3 must be fulfilled by the Security IC Embedded Software. The definition and maintenance of the roles that act on behalf of the functions provided by the hardware must be subject of the Security IC Embedded Software.

The functional requirement FMT_MSA.2 is not included because FMT_MSA.2 requires that “The TSF shall ensure that only secure values are accepted for security attributes.” This is clearly out of scope for the TOE. The design concept of the TOE and the systems in which the TOE is used is based on the fact that the authorized users can protect their data stored by the TOE by using secret keys and a secure access configuration. Therefore the TOE cannot ensure that the security attributes are secure, this is the primary responsibility of the authorized users.

6.3.3 Rationale for the Assurance Requirements

The selection of assurance components is based on the underlying PP [6]. The Security Target uses the same augmentations as the PP. The level EAL4 is chosen in order to meet assurance expectations of digital signature applications and electronic payment systems. Additionally, the requirement of the PP [6] on EAL4 is fulfilled.

The rationale for the augmentations is the same as in the PP. The assurance level EAL4 is an elaborated pre-defined level of the CC, part 3 [3]. The assurance components in an EAL level are chosen in a way that they build a mutually supportive and complete set of components. The requirements chosen for augmentation do not add any dependencies, which are not already fulfilled for the corresponding requirements contained in EAL 4. Therefore, these components add additional assurance to EAL 4, but the mutual support of the requirements is still guaranteed.

As stated in the section 6.3.3 of [6], it has to be assumed that attackers with high attack potential try to attack Security IC used for digital signature applications or payment systems. Therefore specifically AVA_VAN.5 was chosen by the PP [6] in order to assure that even these attackers cannot successfully attack the TOE.

6.3.4 Security Requirements are Internally Consistent

The discussion of security functional requirements and assurance components in the preceding sections has shown that mutual support and consistency are given for both groups of requirements. The arguments given for the fact that the assurance components are adequate for the functionality of the TOE also show that the security functional and assurance requirements support each other and that there are no inconsistencies between these groups.

The Security Functional Requirements must be split in to two categories. The first category is the Security Functionality provided by the hardware platform. This is defined by the Security Functional Requirements defined in the PP [6] and the additional Security Functional Requirements derived from the Organizational Security Policy “P.Add-Components”. The second category is the Security Functionality provided by the Organizational Security Policy “P.DESFire-Emulation”.

The Security Functional Requirements of the PP and the Security Functional Requirements based on P.Add-Components match because the Security Functionality 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 according to FCS_COP.1[HW_DES] and FCS_COP.1[HW_AES] as well as the implementation of the memory partitioning / memory access control and the access control to Special Function Register according to the security functional requirement FDP_ACC.1[MEM] and FDP_ACC.1[SFR] with reference to the Access Control Policies defined in FDP_ACF.1[MEM] and FDP_ACF.1[SFR].

The Security Functionality defined by P.DESFire-Emulation is only provided if the DESFire EV1 Software is called by the Security IC Embedded Software. However to the partitioning of the hardware platform by the MIFARE Firewall the Security Functionality provided by P.DESFire-Emulation cannot be influenced by the Security IC Embedded Software. The partitioning works in both directions so that the Security Functionality of the Security IC Embedded Software cannot be influenced by the DESFire EV1 Software.

A Security IC hardware platform requires Security IC Embedded Software to build a secure product. Thereby the Security IC Embedded Software must support the security functionality of the hardware platform and implement a sufficient management of the security services provided by the hardware. The realization of the Security Functional Requirements within the TOE provides a good balance between flexible configuration and restrictions to ensure a secure behavior of the TOE.

7. TOE Summary Specification

This chapter is composed of sections “Portions of the TOE Security Functionality” and “TOE Summary Specification Rationale”.

7.1 Portions of the TOE Security Functionality

The TOE Security Functionality (TSF) directly corresponds to the TOE security functional requirements defined in chapter 6. The TSF is split into Security Services (SS) and Security Features (SF), which are applicable to phases 4 to 7 of the Security IC product life-cycle.

Note: Parts of the security functionality are configured at the end of phase 3 and all TOE Security Functionality is active after phase 3.

The TOE provides security mechanisms to the Security IC Embedded Software, which are not listed as separated security service in the following. Such mechanisms do not provide a complete portion of TOE Security Functionality, but they can be used to support a portion of security functionality implemented by the Security IC Embedded Software, e.g. the FameXE coprocessor for asymmetric cryptographic algorithms or the CRC calculation for the control of data integrity.

7.1.1 Security Services of the hardware platform

SS.HW_RNG: Random Number Generator

The Random Number Generator continuously produces random numbers with a length of one byte. The TOE implements SS.HW_RNG by means of a physical hardware Random Number Generator working stable within the valid ranges of operating conditions, which are guaranteed by SF.OPC.

The TSF provides a hardware test functionality, which can be used by the Security IC Embedded Software to detect faults in the hardware of the Random Number Generator.

According to “AIS31” [5] the Random Number Generator claims the fulfillment of the requirements of functionality class P2. This means that the Random Number Generator is suitable for generation of signature key pairs, generation of session keys for symmetric encryption mechanisms, random padding bits, zero-knowledge proofs and the generation of seeds for DRNGs.

SS.HW_DES: Triple-DES Coprocessor

The TOE provides the Triple Data Encryption Algorithm (TDEA) according to the Data Encryption Standard (DES). SS.HW_DES is a modular basic cryptographic function, which provides the TDEA algorithm as defined by FIPS PUB 46 by means of a hardware coprocessor and supports (a) the 3-key Triple-DEA algorithm according to keying option 1 and (b) the 2-key Triple DEA algorithm according to keying option 2 in FIPS PUB 46-3 [19]. The two/three 56-bit keys (112-/168-bit) for the 2-key/3-key Triple DES algorithm shall be provided by the Security IC Embedded Software. For encryption the Security IC Embedded Software provides 8 bytes of the plain text and SS.HW_DES calculates 8 bytes cipher text. The calculation output is read by the Security IC Embedded Software. For decryption the Security IC Embedded Software provides 8 bytes of cipher text and SS.HW_DES calculates 8 bytes plain text. The calculation output is read by the Security IC Embedded Software.

SS.HW_AES: AES Coprocessor

The TOE provides the Advanced Encryption Standard (AES) algorithm according to the Advanced Encryption Standard as defined by FIPS PUB 197 [20]. SS.HW_AES is a modular basic cryptographic function, which provides the AES algorithm by means of a hardware coprocessor and supports the AES algorithm with three different key lengths of 128, 192 or 256 bit. The keys for the AES algorithm shall be provided by the Security IC Embedded Software.

The software part of the TOE provides calls of the AES coprocessor with three different key lengths of 128, 192 or 256 bit. The calculation output is read by SS.HW_AES. For decryption the Security IC Embedded Software provides 16 bytes of cipher text. The calculation output is read by SS.HW_AES.

7.1.2 Security Services of the DESFire EV1 Software

SS.DF_AUTH: DESFire Authentication

The TOE provides an authentication mechanism to separate authorized subjects from unauthorized subjects. The authentication of subjects is performed by a cryptographic challenge-response. The TOE supports the cryptographic algorithms 2-key Triple-DES, 3-key Triple-DES and 128-bit AES; for DES according to FIPS PUB 46-3 [19] and for AES according to FIPS PUB 197 [20]. The authentication mechanisms are implemented using the cryptographic coprocessors and the hardware random number generator provided by the hardware platform. The authentication mechanisms are protected against attacks like e.g. replay.

SS.DF_AUTH identifies the user to be authenticated by the currently selected context (specific application, chosen by a ‘select’ command) and the key number indicated in the authentication request. By default and before any authentication request SS.DF_AUTH identifies and authenticates the role Everybody. The roles Administrator, Application Manager, Application User and Originality Key User are authenticated during the authentication request by the knowledge of the respective cryptographic key.

The authentication state is remembered by SS.DF_AUTH and the authentication need not to be performed again as long as none of the following events occur: Issue of a ‘select’ command, occurrence of any error during the processing of commands, change of the key that was used for authentication and reset (any cause, either internal or external reset). These events will reset the authentication state to the default (Everybody). Additionally, if the Application Manager deletes his application the authentication state will be reset as an implication.

Note that the TOE does also allow Single-DES authentication, but this shall not be used in the evaluated product. The TOE supports a backward compatible DES authentication in addition to the standard DES authentication. The backward compatible DES authentication shall not be used in the evaluated product.

SS.DF_ACCESS: Access Control to DESFire Data

SS.DF_ACCESS provides an access control mechanism to the objects and security attributes of the DESFire EV1 Software . The access control mechanism assigns subjects - (possibly multiple) Application Users - to 4 different groups of operations on files. For data files, the operations are “read”, “write”, “read and write” and “change attribute”. For values the operations are “read and decrease”, “read, decrease, limited increase”, “read, decrease, limited increase, increase” and “change attribute”. One subject can be assigned to each group of file operations. The special subjects “Everybody” and “Nobody” can also be assigned.

For applications of the DESFire EV1 Software the operations are “create file” and “delete file”. These operations can be assigned to the Application Manager or to everybody. The assignment is stored in the application attributes. If a file is created the file attributes must be supplied with the create request.

On the DESFire card level the operations are “create application” and “delete application”. These operations can be assigned to the Administrator or to Everybody. The assignment is stored in the card attributes. If an application is created the application attributes must be supplied with the create request. A “delete application” operation will securely delete all application keys by overwriting them with random values.

The Originality Key User is not allowed to perform any action on objects, but with a successful authentication he can prove the authenticity of the Security IC.

SS.DF_ACCESS also controls access to the security attributes and the authentication data. The card attributes and the card master key can only be changed by the Administrator, as long as the Administrator does not freeze the card attributes or freezes the card master key. The application attributes and application master keys can be changed by the Application Manager, as long as the Application Manager does not freeze the application attributes or the application master key. Additionally the Application Manager can change the Application User keys and decide if the Application Users can change their keys or not. For files, the attributes can be changed by the subject that has the “change attribute” right. SS.DF_ACCESS allows the Administrator to specify a default application master key and application keys that will be used when an application is created.

SS.DF_CONFID: DESFire Communication Confidentiality

The TSF SS.DF_CONFID provides a mechanism to protect the communication against eavesdropping. In order to do this the communication can be encrypted. The encryption is performed based on the option in the file attributes. There options can be changed by the file owner (i.e. the subject that has the right to “change attribute” for a file).

The encryption algorithm is the same as the one used during authentication for the session, however SS.DF_CONFID only supports the AES algorithm, therefore it is bound to authentications with this algorithm. Note that the TSF SS.DF_CONFID can be activated after authentication performed with SS.DF_AUTH.

SS.DF_CONFID can also be configured to add data to the unencrypted communication stream that enables the terminal to detect integrity violations, replay attacks or man-in-the-middle attacks.

If an encrypted communication is requested, SS.DF_CONFID also verifies the data sent by the terminal and returns an error code if such an attack is detected. The detection mechanism covers all frames exchanged between the terminal and the DESFire EV1 Software up to the current encrypted frame. Therefore SS.DF_CONFID can detect any injected/modified frame in the communication before the transfer of the encrypted frame, but it cannot detect what frame was injected/modified.

SS.DF_TYPECHECK: DESFire Filetype Consistency Check

SS.DF_TYPECHECK ensures the type consistency of the file types stored by the TOE. For value files the check comprises over- or underflow. Furthermore size limitations of files are obeyed and SS.DF_TYPECHECK ensures that records read/writes are handled specific to the type of the record file.

SS.DF_TRANS: DESFire Transaction Protection

The transaction mechanism implemented by SS.DF_TRANS ensures that either all or none of the (modifying) commands within a transaction are performed. The transaction mechanism is active for backup data files, value files, linear record files and cyclic record files, it is not active for standard data files. All file types with the exception of ‘standard data files’ are called ‘backup files’ in the following.

SS.DF_TRANS is always active for the respective file types. This means that for every modifying operation with a backup file an explicit commit request must be issued in order to let the modifications take effect.

Several reasons will abort a transaction: These are the explicit abort request, chip reset, an authentication request, a ‘select’ command or any failure of a command.

7.1.3 Security Features of the hardware platform

SF.OPC: Control of Operating Conditions

SF.OPC ensures correct operation of the TOE (functions offered by the microcontroller including the standard CPU as well as the Triple-DES coprocessor, AES coprocessor, the arithmetic coprocessor, the memories, registers, I/O interfaces and the other system peripherals) during execution of the IC Dedicated Support Software and Security IC Embedded Software. This includes all specific security mechanisms of the TOE, which are able to provide an active response.

The TOE ensures its correct operation and prevents from any malfunction using the following mechanisms: filtering of power supply and clock frequency input as well as monitoring of voltage supply, clock frequency input and the temperature of the chip by means of sensors. There are multiple sensors for the different ISO/IEC 7816 voltage classes and the contactless operation mode. Light sensors are distributed over the chip surface and used to detect light attacks. Thresholds of the parameters, which are monitored by the mechanisms, are set appropriate to the range where the TOE ensures its correct operation. In addition to the light sensors the EEPROM provides two functions to detect light attacks. The Security IC Embedded Software can select one function and also disable both functions of the EEPROM detection function.

Specific functional units of the TOE are equipped with the Secure Fetch Technology^{TM} or other special circuitry to detect fault injection attacks. These are the Program Counter, the stack pointer, the PSWH register, the MMU address cache registers, the DES coprocessor, AES coprocessor, and the FameXE coprocessor.

If one of the monitored parameters is out of the specified range, either (i) a reset is forced and the actually running program is aborted or (ii) an exception is raised which interrupts the program flow and allows a reaction of the Security IC Embedded Software. A reset is forced by the sensors for voltage, frequency, temperature and light, the Secure Fetch Technology^{TM} and the single fault injection detection in the MMU address cache registers. An exception is forced by the EEPROM light detector and the other single fault injection detection circuitry. In case the TOE resets all components of the hardware platform, they are initialized with their reset values and the TOE provides a reset cause indicator to the Security IC Embedded Software. In the case an exception is raised an indicator for the reason of the exception is provided.

Test Mode is disabled before TOE delivery. The TOE automatically enables the sensors when during startup. Furthermore the TOE defends the sensors from being disabled by the Security IC Embedded Software. The assignment which sensor raises an exception or forces a reset is hard-wired and cannot be changed by the Security IC Embedded Software.

The TOE also controls the specified range of the stack pointer. The stack pointer and the control logic are implemented threefold for the User Mode, System Mode and Super

System Mode (comprising Boot Mode, Test Mode and MIFARE Mode). An exception is generated in case the specified limits are exceeded.

In addition, SF.OPC comprises a sensor, which checks the high voltage of the write process to the EEPROM during each write sequence. The result of this sensor must be read from a Special Function Register and does not force an automatic event (e.g. exception).

SF.PHY: Protection against Physical Manipulation

SF.PHY protects the TOE against manipulation of (i) the IC hardware, (ii) the IC Dedicated Software in ROM, (iii) the Security IC Embedded Software in ROM and EEPROM, (iv) the Application Data in EEPROM and RAM including TSF data in the EEPROM. It also protects User Data and TSF data against disclosure by physical probing when stored or while being processed by the TOE.

The protection of the TOE comprises several security mechanisms in design and construction, which make reverse-engineering and tamper attacks more difficult. These mechanisms comprise dedicated shielding techniques for different components and specific encryption mechanisms for the memories. SF.PHY supports the efficiency of other portions of the security functionality.

SF.PHY also supports the integrity of the EEPROM and the ROM. The EEPROM is able to correct a 1-bit error within each byte. The ROM provides a parity check. The EEPROM corrects errors automatically without user interaction, a ROM parity error forces a reset.

SF.LOG: Logical Protection

SF.LOG implements security mechanisms to limit or eliminate the information in the shape and amplitude of signals or in the time between events, which might be found by measuring such signals. This comprises the power supply and signals on other pads, which are not intentionally used for communication by the terminal, the DESFire EV1 Software or the Security IC Embedded Software. Thereby SF.LOG prevents from disclosure of User Data and TSF data stored and/or processed in the Security IC through measurement of power consumption and subsequent complex signal analysis. This protection of the TOE is enforced by several security mechanisms in the design, which support other portions of security functionality.

The Triple-DES coprocessor includes security mechanisms to prevent SPA/DPA analysis of shape and amplitude of the power consumption and ensures that the calculation time is independent from any key and plain/cipher text.

The AES coprocessor includes security mechanisms to prevent SPA/DPA analysis of shape and amplitude of the power consumption and ensures that the calculation time is with respect to the key length independent from any plain/cipher text.

The FameXE coprocessor provides measures to prevent timing attacks on basic modular function. The calculation time of an operation depends on the lengths of the operands, but not on the value of the operands, with the following exceptions: multiplication with reduction, modular inversion and modular division. These three operations have no constant timing due to correction cycles that are needed based on the calculation method. In addition, mechanisms are included, which provide limitations of the capability for the analysis of shape and amplitude of the power consumption. Of course the FameXE does not realize an algorithm on its own and algorithm-specific leakage countermeasures have to be added by the Security IC Embedded Software when using the FameXE.

Additional security mechanisms being configured by the Security IC Embedded Software comprise (i) FameXE HIGHSEC mode, which adds dummy calculations, and (ii) CPU clock configurations, that can be used to prevent the possibility to synchronize the internal operation with the external clock or to synchronize with the characteristics of the power consumption that can be used as trigger signal to support leakage attacks (DPA or timing attacks)

Some mechanisms described for SF.PHY (e.g. the encryption mechanisms) and for SF.OPC (e.g. the filter mechanisms) also support SF.LOG.

SF.COMP: Protection of Mode Control

SF.COMP provides control of the CPU mode for (i) Boot Mode, (ii) Test Mode and (iii) MIFARE Mode. This includes protection of electronic fuses stored in a protected memory area, the so-called Security Rows, and the possibility to store initialization or pre-personalization data in the so-called FabKey Area.

Control of the CPU mode for Boot Mode, Test Mode and MIFARE Mode prevents from abuse of test functions after TOE delivery. It also inhibits abuse of features, which are used during start-up or reset to configure the TOE. The initial – but not user visible – CPU mode during start-up or reset is the Boot Mode. Hardware circuitry determines whether Test Mode is available or not. If available, the TOE jumps to the IC Dedicated Test Software in Test Mode. Otherwise, the TOE switches to System Mode – the initial user visible CPU mode – and starts execution of the Security IC Embedded Software.

The protection of electronic fuses ensures secure storage of configuration and calibration data, which are set up in Test Mode. SF.COMP protects CPU mode switches regarding Boot Mode, Test Mode and MIFARE Mode in the following way: Switching from Boot Mode to Test Mode or MIFARE Mode is allowed, switching from these modes back to Boot Mode is prevented. Switching to Test Mode is prevented as well after TOE delivery, because Test Mode then is permanently disabled. SF.COMP also ensures that Boot Mode is active only in the boot phase during start-up or reset of the TOE, and cannot be invoked afterwards. Therefore, once the TOE has left the test phase and each time the TOE completed start-up or reset, the MIFARE Mode is the only CPU mode available out of Super System Mode during the normal operation. Super System Mode is indicated by PSWH.SSM being set and means, that one CPU mode out of Boot Mode, Test Mode and MIFARE Mode is active. SF.COMP controls the mode, which is used, when bit PSWH.SSM is set.

The protection of electronic fuses especially ensures that configuration options with regard to the security functionality cannot be changed, abused or influenced in any way. SF.COMP ensures that activation or deactivation of security mechanisms cannot be influenced by the Security IC Embedded Software or the DESFire EV1 Software so that the TSF provides self-protection against interference and tampering by untrusted Security IC Embedded Software or the DESFire EV1 Software.

The TSF controls access to the Security Rows, the top-most 128 Bytes of the EEPROM memory, accessible at reserved addresses in the memory map. The available EEPROM memory space for the Security IC Embedded Software is reduced by this area. SF.COMP provides three memory areas in the Security Rows that can be used by the Security IC Embedded Software. These are

- the User Read Only Area
- the User Write Protected Area and
- the User Write Once Area.

The User Read Only Area contains 32 bytes, which are read-only for the Security IC Embedded Software. The User Write Protected area contains 16 bytes, which can be write-protected by the Security IC Embedded Software on demand. The User Write Once Area contains 32 bytes of which each bit can separately be set to ‘1’ once only, and not reset to ‘0’.

If the Card Disable Function is used (refer to section 1.4.1.2) SF.COMP inhibits any start-up of the Security IC Embedded Software once the Security IC Embedded Software disables the Security IC.

SF.COMP also provides the FabKey Area in which pre-personalization data and identification data can be stored. The FabKey area does not belong to the Security Rows and is not protected by hardware mechanisms. The FabKey Area as well as the Security Rows can be used by SF.COMP to store a unique identification for each die.

For all areas the initial values are set during chip testing and pre-personalization. They depend on the choice of the Security IC Embedded Software developer and are included in the Order Entry Form. The User Write Protected Area and the User Write Once Area are designed to store the identification of a (fully personalized) Security IC (e.g. smartcard) or a sequence of events over the life cycle, that can be coded by an increasing number of bits set to "one" or protecting bytes, respectively.

SF.COMP limits the capabilities of the test functions and provides test personnel during phase 3 with the capability to store identification and/or pre-personalization data and/or supplements of the Security IC Embedded Software in the EEPROM. SF.COMP provides self-protection against interference and tampering by untrusted subjects both in the Test Mode and in the other modes. It also enforces the separation of domains regarding the IC Dedicated Software and the Security IC Embedded Software.

SF.MEM_ACC: Memory Access Control

SF.MEM_ACC controls access of any subject (program code comprising processor instructions) to the memories of the TOE through the Memory Management Unit. Memory access is based on virtual addresses that are mapped to physical addresses. The CPU always uses virtual addresses. The Memory Management Unit performs the translation from virtual to physical addresses and the physical addresses are passed from the Memory Management Unit to the memory interfaces to access the memories. The access control is performed in two ways:

- Memory partitioning: Each memory type ROM, RAM and EEPROM is partitioned into two parts. In Boot Mode, MIFARE Mode, System Mode and User Mode the CPU has access to only one part of each memory type. Access to both parts of each type is allowed in Test Mode for testing.
- Memory segmentation in User Mode: The three accessible parts of the memory in ROM, RAM and EEPROM can be segmented into smaller areas. Access rights (readable, writeable or executable) can be defined for these segments. In addition, access rights to Special Function Registers related to hardware components can be defined for code that is executed in a segment.

Memory partitioning is fixed and cannot be changed. It is determined during production of the TOE and is solely dependent on the MIFARE configuration.

Memory segmentation can be defined in System Mode. The segmentation is active when the CPU switches to User Mode. The segments, their access rights and the access rights to Special Function Registers related to hardware components are defined in the MMU Segment Table. The MMU Segment Table stores five values for each segment: The memory access rights, the virtual start address of the segment, the virtual end address of the segment, the address offset for the segment and the access rights to Special Function Registers for code that is executed in the segment. The address offset is used to relocate the segment anywhere in the memory map. The resulting address computed by the Memory Management Unit cannot overrule memory partitioning. Up to 64 segments can be defined in the MMU Segment Table. Special configurations of the memory access rights allow to specify less than 64 segments and to split the MMU Segment Table into several parts being stored at different locations in memory.

Note that the MMU Segment Table itself is stored in the memory and therefore the table itself can be placed in a segment accessible in User Mode.

In addition, SF.MEM_ACC permanently checks whether accessed addresses point to physically implemented memory. Access to forbidden memory addresses in User Mode and accesses outside the boundaries of the physical memory are notified by raising an exception.

As stated above the Memory Management Unit handles access rights to Special Function Registers related to hardware components for code running in User Mode. This information is used by SF.SFR_ACC to grant or block a certain access. The access rights can be defined for up to 16 groups of Special Function Registers, which are related to 16 hardware components. SF.SFR_ACC receives the access rights to these 16 groups from the Memory Management Unit as well for the other CPU modes. In Boot Mode, Test Mode and System Mode the Memory Management Unit indicates full access to the 16 groups. In MIFARE Mode the Memory Management Unit indicates access rights as set in two Special Function Registers, which cannot be modified in MIFARE Mode. Thus, MIFARE Mode can be restricted in its access to the 16 hardware components on demand of the Security IC Embedded Software. Note that SF.MEM_ACC only provides the access rights to SF.SFR_ACC, the access control is enforced by SF.SFR_ACC.

SF.SFR_ACC: Special Function Register Access Control

SF.SFR_ACC controls access to the Special Function Registers and CPU mode switches based on Special Function Register PSWH.

SF.SFR_ACC implements access control to the Special Function Registers as specified in the Access Control Policy and the Security Functional Requirements FDP_ACC.1[SFR] and FDP_ACF.1[SFR].

The function of the Special Function Register and the CPU mode determine, whether read and/or write access to a Special Function Register is allowed or not. For example, read access to the DES key register is write-only according to its function and its write access is granted depending on the CPU mode. Similar for the output register of the Random Number Generator, which is read-only based on its function, and read access is granted based on the CPU mode.

SF.SFR_ACC ignores accesses to Special Function Registers, which are not allowed. Ignoring means that a write access has no influence and/or a read access always returns a fixed value independent of the true value of the Special Function Register.

Some Special Function Registers are implemented threefold, one for User Mode, a second one for System Mode and a third one for Super System Mode meaning Boot Mode, Test Mode and MIFARE Mode. Hence, such Special Function Registers are already separated from the outset.

SF.SFR_ACC relies on access rights to Special Function Registers related to hardware components, which are provided by SF.MEM_ACC. Access rights to all other Special Function Registers are pre-defined and cannot be changed.

This implies that code running in User Mode or MIFARE Mode is not able to use SS.HW_RNG, SS.HW_DES, and SS.HW_AES until access to the respective group of Special Function Registers is explicitly granted by code running in System Mode. This holds for all 16 hardware components, which are controlled by the 16 groups of Special Function Registers related to hardware components.

SF.SFR_ACC also implements transitions among CPU modes based on Special Function Register PSWH. This Special Function Register contains two bits, which are PSWH.SSM and PSWH.SM. Bit PSWH.SSM indicates Super System Mode when set, which means, that one out of Boot Mode, Test Mode and MIFARE Mode is active. Bit PSWH.SM indicates System Mode when set. If both bits are zero, the CPU operates in User Mode.

The CPU mode changes by the following operations.

- Call of a system call vector (SVEC) address or a configuration vector (CVEC) address. A call of a SVEC sets bit PSWH.SM and enables System Mode, a call of a CVEC sets bit PSWH.SSM and enables MIFARE Mode. Calls of SVEC addresses are only allowed in User Mode, otherwise an exception is raised.
- Execution of an exception or interrupt. Any event that leads to the execution of an exception sets bit PSWH.SM. Interrupts can be executed in User Mode or in System Mode. The Security IC Embedded Software running in System Mode can configure this option at run time and based on this configuration bit PSWH.SM is modified or not.
- Return from an exception/interrupt or vector call with a RETI instruction. This restores the value of PSWH to the value before the event occurred. A RETI in User Mode is allowed only in case interrupts are allowed to be executed in User Mode and an interrupt is actually active, otherwise an exception is raised.
- Execution of an LCALL/ACALL/ECALL instruction to a specific address. A call of address 0x800000 in System Mode enables User Mode and starts execution at this (virtual) address. This is similar to a CVEC or SVEC call, but no return address is pushed onto the stack.
- Direct modification of the two bits in PSWH. Hardware provided by SF.SFR_ACC ensures that the bits can only be cleared. Therefore it is not possible for code running in User Mode to enter System Mode, but System Mode can switch to User Mode.

Two CPU modes are available to the Security IC Embedded Software, which are System Mode and User Mode. System Mode is the more privileged CPU mode since it allows access to all Special Function Registers of the hardware components and for system management (i.e. configuration of Memory Management Unit, clock settings and some mechanisms provided by SF.LOG). User Mode is the less privileged, but with regard to the hardware components it can be made as powerful as System Mode.

SF.SFR_ACC and SF.COMP together ensure that other CPU modes are not available to the Security IC Embedded Software, but reserved for specific purposes fulfilled by the IC Dedicated Software. In addition, SF.MEM_ACC provides separation of the memories and access control information.

7.2 TOE Summary Specification Rationale

7.2.1 Rationale for the portions of the TOE security functionality

The following table provides a mapping of portions of the TOE security functionality to the Security Functional Requirements. The mapping is described in detail in the text following the table (only in the full version of the Security Target).

Table 18: Mapping of Security Functional Requirements and the portions of the TOE Security Functionality

	SS.HW_RNG	SS.HW_DES	SS.HW_AES	SS.DF_AUTH	SS.DF_ACCESS	SS.DF_CONFID	SS.DF_TYPECHECK	SS.DF_TRANS	SF.OPC	SF.PHY	SF.LOG	SF.COMP	SF.MEM_ACC	SF.SFR_ACC
FAU_SAS.1												X		
FCS_RNG.1	X													
FDP_IFC.1	X	X	X	X		X			X		X			
FDP_ITT.1	X	X	X	X		X			X		X			
FPT_ITT.1	X	X	X	X		X			X		X			
FMT_LIM.1												X	X	X
FMT_LIM.2												X	X	X
FPT_PHP.3										X				
FPT_FLS.1		X	X	X					X					
FRU_FLT.2									X					
FCS_COP.1[HW_DES]		X												
FCS_COP.1[HW_AES]			X											
FDP_ACC.1[MEM]													X	
FDP_ACC.1[SFR]														X
FDP_ACF.1[MEM]													X	
FDP_ACF.1[SFR]														X
FMT_MSA.1[MEM]													X	
FMT_MSA.1[SFR]														X
FMT_MSA.3[MEM]													X	
FMT_MSA.3[SFR]														X
FMT_SMF.1[HW]													X	X

	SS.HW_RNG	SS.HW_DES	SS.HW_AES	SS.DF_AUTH	SS.DF_ACCESS	SS.DF_CONFID	SS.DF_TYPECHECK	SS.DF_TRANS	SF.OPC	SF.PHY	SF.LOG	SF.COMP	SF.MEM_ACC	SF.SFR_ACC
FMT_SMR.1[DESFIRE]				X	X									
FDP_ACC.1[DESFIRE]					X									
FDP_ACF.1[DESFIRE]					X									
FMT_MSA.1[DESFIRE]					X									
FMT_MSA.3[DESFIRE]					X									
FMT_SMF.1[DESFIRE]				X	X									
FDP_ITC.2				X	X									
FPT_TDC.1							X							
FIA_UID.2				X										
FIA_UAU.2				X										
FIA_UAU.5	X	X	X	X										
FMT_MDT.1				X										
FTP_TRP.1			X			X								
FCS_CKM.4				X	X									
FDP_ROL.1								X						
FPT_RPL.1	X	X	X	X		X								

An "X" in the above table means that the specific portion of the TOE security functionality realizes or supports the functionality required by the respective Security Functional Requirement.

8. Annexes

8.1 Further Information contained in the PP

Chapter 7 of the PP “Security IC Platform Protection Profile” [6] provides further information. Section 7.1 in [6] describes the development and production process of Security ICs including a detailed life-cycle description and a description of the assets of the IC Designer/Manufacturer. Section 7.2 in [6] comprises security aspects of the Security IC Embedded Software, i.e. further information regarding A.Resp-Appl and examples of specific Functional Requirements for the Security IC Embedded Software. Section 7.3 in [6] gives examples of Attack Scenarios.

8.2 Glossary and Vocabulary

Administrator

(in the sense of the Common Criteria) The TOE may provide security functionality which can or need to be administrated (i) by the Security IC Embedded Software or (ii) using services of the TOE after delivery to Phases 4-6. Then a privileged user (in the sense of the Common Criteria, refer to definition below) becomes an administrator.

Application Data

All data managed by the Security IC Embedded Software in the application context. Application data comprise all data in the final Security IC.

Boot Mode

CPU mode of the TOE dedicated to start-up and reset of the TOE. This mode is not accessible for the Security IC Embedded Software.

Composite Product

Integrator Role installing or finalizing the IC Embedded Software and the applications on platform transforming the TOE into the impersonalized Composite Product after TOE delivery

The TOE Manufacturer may implement IC Embedded Software delivered by the Security IC Embedded Software Developer before TOE delivery (e.g. if the IC Embedded Software is implemented in ROM or is stored in the non-volatile memory as service provided by the IC Manufacturer or IC Packaging Manufacturer).

Composite Product Manufacturer

The Composite Product Manufacturer has the following roles (i) the Security IC Embedded Software Developer (Phase 1), (ii) the Composite Product Integrator (Phase 5) and (iii) the Personalize (Phase 6). If the TOE is delivered after Phase 3 in form of wafers or sawn wafers (dice) he has the role of the IC Packaging Manufacturer (Phase 4) in addition.

The customer of the TOE Manufacturer, who receives the TOE during TOE Delivery. The Composite Product Manufacturer includes the Security IC Embedded Software developer and all roles after

TOE Delivery up to Phase 6 (refer to [6], Figure 2 on page 240H10 and Section 7.1.1)

CPU mode

Mode in which the CPU operates. The TOE supports five CPU modes, which are Boot Mode, Test Mode, MIFARE Mode, System Mode and User Mode.

exception interrupt

Non-maskable interrupt of program execution jumping to fixed addresses (depending on the exception source) and enabling System Mode. Sources of exceptions are hardware breakpoints, single fault injection detections, illegal instructions, stack overflows, unauthorized system call vector calls, execution of RETI instruction in User Mode, and the MMU exceptions access violation and access collision.

FabKey Area

A memory area in the EEPROM containing data, which are programmed during testing by the IC Manufacturer. The amount of data and the type of information can be selected by the customer.

IC Dedicated Software

IC proprietary software embedded in a Security IC (also known as IC firmware) and developed by the IC Developer. Such software is required for testing purpose (IC Dedicated Test Software) but may provide additional services to facilitate usage of the hardware and/or to provide additional services (IC Dedicated Support Software).

IC Dedicated Support Software

That part of the IC Dedicated Software (refer to above) which provides functions after TOE Delivery. The usage of parts of the IC Dedicated Software might be restricted to certain phases.

IC Dedicated Test Software

That part of the IC Dedicated Software (refer to above) which is used to test the TOE before TOE Delivery but which does not provide any functionality thereafter.

End-consumer

User of the Composite Product in Phase 7

Initialization Data

Pre-personalization Data defined by the Composite Product Manufacturer to setup the internal data structures and possible supplements of the Security IC Embedded Software and related data to allow further life-cycle changes based on this data.

Any data supplied by the Composite Product Manufacturer that is injected into the non-volatile memory by the Card manufacturer (Phase 5).

Integrated Circuit (IC)

Electronic component(s) designed to perform processing and/or memory functions.

Memory

IC hardware component, that stores code and/or data, i.e. ROM, RAM or EEPROM of the TOE.

Memory Management Unit

The MMU maps the virtual addresses used by the CPU into the physical addresses of RAM, ROM and EEPROM. This mapping is done based on (a) memory partitioning and (b) memory segments for code running in User Mode. Memory partitioning is fixed, whereas up to 64 memory segments can be configured individually. Each segment can be (i) positioned and sized (ii) enabled and disabled, (iii) configured for access rights in terms of read, write and execute in User Mode and (iv) configured for User Mode access rights to Special Function Registers related to hardware components of code executed in this segment.

Memory Segment

Address space provided by the Memory Management Unit according to the configuration in the MMU Segment Table. A memory segment defines a memory area, are accessible for code running in User Mode. Memory segments may address RAM, ROM and EEPROM.

MIFARE

Contactless smartcard interface standard complying with ISO/IEC 14443 A.

MIFARE Mode

CPU mode of the TOE dedicated to execution of the DESFire EV1 Software, which is part of the IC Dedicated Support Software. This mode is not accessible for the Security IC Embedded Software.

MMU Segment Table

This structure defines the memory segments for code running in User Mode, which are controlled by the MMU. The structure can be located anywhere in the memory that is available in System Mode. It also contains User Mode access rights to Special Function Registers related to hardware components of code executed in each segment.

Pre-personalization Data

Pre-personalization Data defined by the TOE Manufacturer to identify the TOE and to keep track of the Security IC's production and further life-cycle phases are considered as belonging to the TSF data. These data are for instance used for traceability and for TOE identification (identification data). It can also be used to protect the transport and support an originality check of the hardware platform.

S^{2}C

Smart card interface standard complying with ISO/IEC 18092.

Security IC

(as used in the PP [6]) Composition of TOE, Security IC Embedded Software, User Data and package (Security IC carrier).

Security IC Embedded Software

Software embedded in a Security IC and normally not being developed by the IC Designer. The Security IC Embedded Software is designed in Phase 1 and embedded into the Security IC in Phase 3 or in later phases of the Security IC product life-cycle.

Some part of that software may actually implement a Security IC application others may provide standard services. Nevertheless, this distinction does not matter here so that the Security IC Embedded Software can be considered as being application dependent whereas the IC Dedicated Software is definitely not.

Security IC Product

Composite product which includes the Security Integrated Circuit (i.e. the TOE) and the Embedded Software and is evaluated as composite target of evaluation in the sense of the Supporting Document

Security Rows

Top-most 128 bytes of the EEPROM memory reserved for configuration purposes as well as dedicated memory area for the Security IC Embedded Software to store life-cycle information about the TOE.

Special Function Registers

Registers used to access and configure the functions for communication with an external interface device, the cryptographic coprocessors for Triple-DES or AES, the FameXE coprocessor for basic arithmetic functions to perform asymmetric cryptographic algorithms, the random numbers generator and chip configuration.

Super System Mode

This term represents either Boot Mode, Test Mode or MIFARE Mode.

System Mode

CPU mode of the TOE with unrestricted access to the hardware resources. The Memory Management Unit can be configured in System Mode.

Test Features

All features and functions (implemented by the IC Dedicated Test Software and/or hardware) which are designed to be used before TOE Delivery only and delivered as part of the TOE.

Test Mode

CPU mode of the TOE for its configuration and execution of the IC Dedicated Test Software. The Test Mode is permanently and irreversible disabled after production testing. Specific Special Function Registers are accessible in Test Mode for test purposes.

TOE Delivery

The period when the TOE is delivered which is (refer to [6], Figure 2 on page (10) either (i) after Phase 3 (or before Phase 4) if the TOE is delivered in form of wafers or sawn wafers (dice) or (ii) after Phase 4 (or before Phase 5) if the TOE is delivered in form of packaged products.

TOE Manufacturer — The TOE Manufacturer must ensure that all requirements for the TOE (as defined in [6], Section 1.2.2) and its development and production environment are fulfilled (refer to [6], Figure 2 on page 10).

The TOE Manufacturer has the following roles: (i) IC Developer (Phase 2) and (ii) IC Manufacturer (Phase 3). If the TOE is delivered after Phase 4 in form of packaged products, he has the role of the (iii) IC Packaging Manufacturer (Phase 4) in addition.

TSF data — Data created by and for the TOE, which might affect the operation of the TOE. This includes information about the TOE’s configuration, if any is coded in non-volatile non-programmable memories (ROM), in specific circuitry, in non-volatile programmable memories (for instance EEPROM) or a combination thereof.

User — (in the sense of the Common Criteria) The TOE serves as a platform for the Security IC Embedded Software. Therefore, the “user” of the TOE (as used in the Common Criteria assurance class AGD: guidance) is the Security IC Embedded Software. Guidance is given for the Security IC Embedded Software Developer.

On the other hand the Security IC (with the TOE as a major element) is used in a terminal where communication is performed through the ISO/IEC interface provided by the TOE. Therefore, another “user” of the TOE is the terminal (with its software).

User Data — All data managed by the Security IC Embedded Software in the application context. User data comprise all data in the final Security IC except the TSF data.

User Mode — CPU mode of the TOE. Access to memories is controlled by the Memory Management Unit. Access to Special Function Registers is restricted.

8.3 List of Abbreviations

CC	Common Criteria Version 3.1
CIU	Contactless Interface Unit
CPU	Central Processing Unit
DEA	Data Encryption Algorithm
DES	Data Encryption Standard
DRNG	Deterministic Random Number Generator
EAL	Evaluation Assurance Level
ECC	Elliptic Curve Cryptography

IC	Integrated circuit
IT	Information Technology
MMU	Memory Management Unit
MX	Memory eXtension
NDA	Non Disclosure Agreement
NFC	Near Field Communication
PKC	Public Key Cryptography
PP	Protection Profile
PSWH	Program Status Word (High byte)
SAR	Security Assurance Requirement
SFR	as abbreviation of the CC term: Security Functional Requirement, as abbreviation of the technical term of the SmartMX-family: Special Function Register[81]
SIM	Subscriber Identity Module
SOF	Strength of Function
SF	Security Feature
SS	Security Service
ST	Security Target
TOE	Target of Evaluation
TRNG	True Random Number Generator
TSF	TOE Security Functionality
TSFI	TSF Interface
TSP	TOE Security Policy
UART	Universal Asynchronous Receiver and Transmitter

[81] To avoid confusion this Security Target does not use SFR as abbreviation for Special Function Register in the explanatory text. However, the abbreviation is used in objective or security functionality identifiers and to distinguish iterations.

9. Bibliography

9.1.1 Evaluation Documents

[1] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, Version 3.1, Revision 3, July 2009, CCMB-2009-07-001

[2] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements, Version 3.1, Revision 3, July 2009, CCMB-2009-07-002

[3] Common Criteria, Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements, Version 3.1, Revision 3, July 2009, CCMB-2009-07-003

[4] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1, Revision 3, July 2009, CCMB-2009-07-004

[5] Anwendungshinweise und Interpretationen zum Schema, AIS31: Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren, Version 1, 25.09.2001, Bundesamt für Sicherheit in der Informationstechnik

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

[7] Supporting Document Mandatory Technical Document Application of Attack Potential to Smartcards, Version 2.7, Revision 1, March 2009, CCDB-2009-03-001

9.1.2 Developer Documents

[8] Product data sheet P5CD016/021/041/051 and P5Cx081 family; Secure dual interface and contact PKI smart card controller, NXP Semiconductors, Business Line Identification

[9] Instruction Set, SmartMX-Family, Secure and PKI Smart Card Controller, Philips Semiconductors

[10] Guidance, Delivery and Operation Manual NXP Secure Smart Card Controllers P5CD016V1D/P5CD021V1D/P5CD041V1D/P5Cx081V1D, NXP Semiconductors, Business Line Identification

[11] MIFARE DESFire Functional Specification, NXP Semiconductors, Business Line Identification

[12] Order Entry Form P5CD016, NXP Semiconductors, Business Line Identification

[13] Order Entry Form P5CD021, NXP Semiconductors, Business Line Identification

[14] Order Entry Form P5CD041, NXP Semiconductors, Business Line Identification

[15] Order Entry Form P5CD081, NXP Semiconductors, Business Line Identification

[16] Order Entry Form P5CN081, NXP Semiconductors, Business Line Identification

[17] NXP Secure Smart Card Controllers P5CD016/021/041V1A and P5Cx081V1A Security Target Lite, Revision 0.1 NXP Semiconductors, 10 August 2009

[18] Data Sheet P5CD016/021/041 V1D and P5Cx081 V1D with MIFARE DESFire EV1 OS, NXP Semiconductors, Business Line Identification

9.1.3 Other Documents

[19] FIPS PUB 46-3 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION DATA ENCRYPTION STANDARD (DES) Reaffirmed 1999 October 25

[20] FIPS PUB 197 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, ADVANCED ENCRYPTION STANDARD (AES), National Institute of Standards and Technology, 2001 November 26

[21] PKCS #1: RSA Cryptography Specifications, Version 2.0. RSA Laboratories, September 1998

[22] ISO/IEC 7816-2:1996 Information technology – Identification cards – Integrated circuit(s) cards with contacts – Part 2: Dimensions and location of contacts

[23] ISO/IEC 7816-3:1997 Information technology – Identification cards – Integrated circuit(s) cards with contacts – Part 3: Electronic signals and transmission protocols

[24] ISO/IEC 14443-3:2001 Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 3: Initialization and anticollision

[25] ISO/IEC 14443-4:2001 Identification cards – Contactless integrated circuit(s) cards – Proximity cards – Part 4: Transmission protocol

[26] ISO/IEC 18092:2004: Information technology – Telecommunications and information exchange between systems – Near Field Communication – Interface and Protocol (NFCIP-1)

[27] ISO/IEC 9797-1: Information *technology – Security techniques – Message Authentication* – Part 1: Mechanisms using a block cipher, 1999

[28] FIPS PUB 81 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, DES modes of operation, US Department of Commerce/National Institute of Standards and Technology, 1980 December 2

[29] ANSI X9.52, American National Standard: Triple data encryption algorithm modes of operation, 1998 November 9

10. Legal information

10.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

10.2 Disclaimers

General — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in medical, military, aircraft, space or life support equipment, nor in applications where failure or malfunction of a NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is for the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

10.3 Licenses

ICs with DPA Countermeasures functionality

NXP ICs containing functionality implementing countermeasures to Differential Power Analysis and Simple Power Analysis are produced and sold under applicable license from Cryptography Research, Inc.

10.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.

<Patent ID> — owned by <Company name>

10.5 Trademarks

Notice: All referenced brands, product names, service names and trademarks are property of their respective owners.

FabKey — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

11. Contents

Please be aware that important notices concerning this document and the product(s) described herein, have been included in the section 'Legal information'.

For more information, please visit: http://www.nxp.com
For sales office addresses, email to: salesaddresses@nxp.com

Date of release: 24 October 2011