NXP Secure Smart Card Controller P60D024/016/012PVB/PVB(Y)

Security Target Lite

Rev. 2.2 — 04 November 2013
BSI-DSZ-CC-0810

Evaluation documentation
Public

Document information

Info	Content
Keywords	CC, Security Target Lite, P60D024/016/012PVB, P60D024/016/012PVB(Y)
Abstract	Security Target Lite of the NXP Secure Smart Card Controller P60D024/016/012PVB/PVB(Y), 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 6 augmented.

Revision history

Rev	Date	Description
Rev. 2.2	04 November 2013	derived from Security Target P60D024/016/012PVB/PVB(Y), Rev. 2.2

Latest version is: Rev. 2.2 (04 November 2013)

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 Controller P60D024/016/012PVB/PVB(Y) Security Target Lite, NXP Semiconductors, Business Unit Identification, Rev. 2.2, 04 November 2013”

1.2 TOE Reference

The TOE is named “NXP Secure Smart Card Controller P60D024/016/012PVB/PVB(Y) with IC Dedicated Software”.

In this document the TOE is abbreviated to NXP Secure Smart Card Controller P60D024/016/012PVB.

1.3 TOE Overview

1.3.1 Usage and major security functionality of the TOE

The TOE is the IC hardware platform NXP Secure Smart Card Controller P60D024/016/012PVB with IC Dedicated Software and documentation describing the Instruction Set and the usage. The TOE is delivered as with a customer specific Security IC Embedded Software.

The IC hardware platform NXP Secure Smart Card Controller P60D024/016/012PVB is a microcontroller incorporating a central processing unit, memories accessible via a Memory Management Unit, cryptographic coprocessors, other security components and two communication interfaces. The central processing unit supports a 32-/24-/16-/8-bit instruction set optimized for smart card applications, which is a super set of the 80C51 family instruction set. The first and in some cases the second byte of an instruction are used for operation encoding. On-chip memories are ROM, RAM and EEPROM. The non-volatile EEPROM can be used as data or program memory. It consists of high reliable memory cells, which guarantee data integrity. The EEPROM is optimized for applications requiring reliable non-volatile data storage for data and program code. EEPROM double read function is included for correct memory readout. Dedicated security functionality protects the contents of all memories.

The IC Dedicated Software comprises IC Dedicated Test Software for test purposes and IC Dedicated Support Software. The IC Dedicated Support Software consists of Boot-ROM Software controlling the boot process of the hardware platform and Firmware Operating System which can be called by the Security IC Embedded Software. The Firmware Operating System provides an interface for programming of the internal EEPROM memory, which is mandatory for use by the Security IC Embedded Software when programming the EEPROM memory.

The documentation includes a Data Sheet, a description of the Instruction Set, a Guidance Document and a document describing the delivery of the product. This documentation comprises a description of the architecture, the secure configuration and usage of the IC hardware platform and the IC Dedicated Firmware by the Security IC Embedded Software.

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 exceptions by the Security IC Embedded Software. The different CPU modes and the Memory Management Unit support the implementation of multi-application projects using the NXP Secure Smart Card Controller P60D024/016/012PVB.

A Security IC must provide high security in particular when being used in the banking and finance market, in electronic commerce or in governmental applications because the TOE is intended to be used in a potential insecure environment. Hence the TOE shall maintain

- the integrity and the confidentiality of code and data stored in its memories and
- the different CPU modes with the related capabilities for configuration and memory access and
- 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.

NXP Secure Smart Card Controller P60D024/016/012PVB basically provides a hardware platform for an implementation of a smart card application with

- 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 are suitable for public key cryptography and elliptic curve cryptography,
- a True Random Number Generator,
- memory management control,
- cyclic redundancy check (CRC) calculation,
- ISO/IEC 7816 contact interface with UART,
- ISO/IEC 14443 A contactless interface.

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 security exceptions as well as sensors, which allow operation under specified conditions only. Memory encryption is used for memory protection and chip shielding is added to the chip.

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 the 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 NXP Secure Smart Card Controller P60D024/016/012PVB is provided as IC hardware platform for various operating systems and applications with high security requirements.

1.3.3 Required non-TOE hardware/software/firmware

None

1.4 TOE Description

1.4.1 Physical Scope of TOE

The NXP Secure Smart Card Controller P60D024/016/012PVB is manufactured in an advanced 90nm CMOS technology. A block diagram of the IC is depicted in Fig 1.

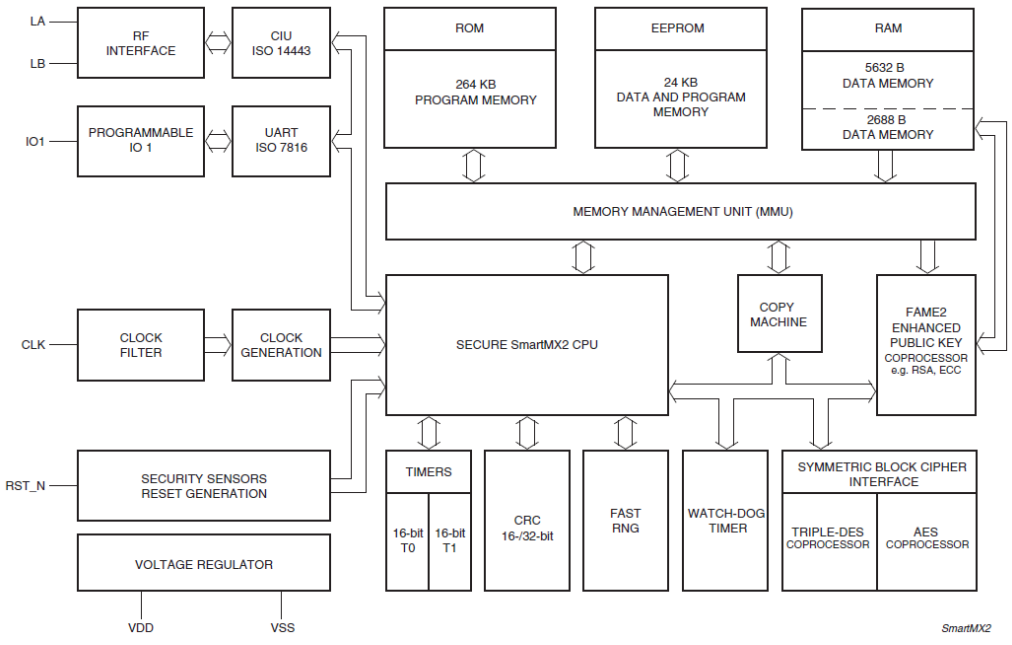

Fig 1. Block Diagram of NXP Secure Smart Card Controller P60D024/016/012PVB

The TOE consists of the IC hardware platform and Security IC Dedicated Software as composed of Security IC Dedicated Test Software and Security IC Dedicated Support Software. All other software is called Security IC Embedded Software. The Security IC Embedded Software is not part of the TOE. The TOE components are listed in Table 1.

1.4.1.1 TOE components

Table 1. Components of the TOE

Type	Name	Release	Date	Form of delivery
IC Hardware	NXP Secure Smart Card Controller P60D024/016/012PVB	VB	20 September 2011	wafer, module, inlay, package (dice have nameplate 9047A)[1]
	NXP Secure Smart Card Controller P60D024/016/012PVB(Y)	VB(Y)	12 September 2012	wafer, module, inlay, package (dice have nameplate 9047A)[1]
Security IC Dedicated Test Software	Test-ROM Software	08.07	21 September 2011	Test-ROM on the chip acc. to 9047A_BG002_TESTROM_v1_btos_08v07_fos_5v0.hex
Security IC Dedicated Support Software	Boot-ROM Software	08.07	21 September 2011	Boot-ROM on the chip acc. to 9047A_BG002_TESTROM_v1_btos_08v07_fos_5v0.hex
	Firmware Operating System (FOS)	5.00	21 September 2011	Firmware Operating System on the chip acc. to 9047A_BG002_TESTROM_v1_btos_08v07_fos_5v0.hex
Document	Product Data Sheet SmartMX2 family P60D012/016/024, Secure high-performance smart card controller			Electronic Document
Document	Instruction Set for the SmartMX2 family, Secure high-performance smart card controller			Electronic Document
Document	NXP Secure Smart Card Controller P60D024/016/012 VB Guidance and Operation Manual			Electronic Document
Document	SmartMX2 family P60D024/016/012 VB Wafer and delivery specification			Electronic Document

[1] ROM Code numbers 001 to 004 and 096 to 099 are assigned to Version VB, and ROM Code numbers 005 to 095 are assigned to version VB(Y). Note that modification according IAR [18] is included in all devices that have a ROM code number *xxx* in the range of 005 to 095. The ROM code number can unambiguously identified by reading Security Row bytes RCN0 (DFFF8A) and RCN1 (DFF8B), see [9]. In addition the ROM Code number is physically visible on the dice surface according to [12].

Type	Name	Release	Date	Form of delivery
Document	Product data sheet addendum: SmartMX2 family, Post Delivery Configuration (PDC)			Electronic Document
Document	Product data sheet addendum: SmartMX2 family, Chip Health Mode (CHM)			Electronic Document

The TOE contains a Security IC Dedicated Software which consists of a Security IC Dedicated Test Software and Security IC Dedicated Support Software. The Security IC Dedicated Test Software contains the Test-ROM Software; the Security IC Dedicated Support Software contains the Boot-ROM Software and the Firmware Operating System.

The version of the Security IC Dedicated Software specified in Table 1 can be identified by the Security IC Embedded Software by reading out the ROM Code Number (RCN) as defined in [9], section 31.2.1. Furthermore the version of the Firmware Operating System as part of the Security IC Dedicated Support Software can be read-out by Security IC Embedded Software using FVEC interface as specified in [9], section 13.4.1 Emulation Control Interface (FVEC0).

1.4.2 Evaluated configurations

The customer can select different configurations of the NXP Secure Smart Card Controller P60D024/016/012PVB. The configuration options are structured as major and minor configuration options.

The TOE can be delivered with specific configurations that are named P60D024PVB, P60D016PVB or P60D012PVB each with the same IC Dedicated Software. In short form the TOE is named P60D024/016/012PVB. "024", "016" and "012" specify the accessible EEPROM memory.

1.4.2.1 Major configuration options

Three major configurations are present, which are denoted by the names P60D024PVB, P60D016PVB and P60D012PVB. All of them are equipped with an EEPROM of 24 kBytes and both, the ISO/IEC 7816 contact interface and the ISO/IEC 14443 contactless interface. Their major differences are related to the availability of EEPROM space as detailed below.

Each major configuration is provided with several minor configuration options, which are introduced in Section 1.4.2.2. Each major configuration also provides customers with several options for reconfiguration (Post Delivery Configuration), which are described in Section 1.4.2.3 in detail.

Table 2. Evaluated major configuration options

Major configuration	P60D024PVB	P60D016PVB	P60D012PVB
available EEPROM memory to the Security IC Embedded Software	24 kBytes except for 512 Bytes reserved for Security Rows and configuration data of the manufacturer and 768 Bytes reserved for IC Dedicated Support Software (Firmware OS)	16 kBytes except for 512 Bytes reserved for Security Rows and configuration data of the manufacturer and 768 Bytes reserved for IC Dedicated Support Software (Firmware OS)	12 kBytes except for 512 Bytes reserved for Security Rows and configuration data of the manufacturer and 768 Bytes reserved for IC Dedicated Support Software (Firmware OS

The evaluated major configuration ‘P’ as well as not evaluated major configuration options ‘M’, and ‘D’ for OS emulations can be selected via Order Entry Forms [13], [14] respectively [15], which are individual for each type name.

Table 3. Evaluated major configuration options

Name	Value	Description
OS Emulation	• P: NO OS Emulation	MIFARE Plus and/or MIFARE DESFire Emulation. Major configuration options • M: MIFARE Plus emulation • D: MIFARE DESFire Emulation and related memory image selection in EEPROM for data storage of the MIFARE Plus or MIFARE DESFire emulation data • M2: 2kByte • M4: 4kByte • D2: 2kByte • D4: 4kByte • D8: 8kByte are not evaluated major configurations option.

1.4.2.2 Minor configuration options

Minor configuration options can be selected by the customer via Order Entry Forms [13], [14] respectively [15], which are individual for each type name. The first seven characters in the name of a major configuration give the type name and therewith the Order Entry Form belonging to. The Order Entry Form identifies the minor configuration options, which are supported by a major configuration out of those introduced in Table 2 and Table 3.

Table 4. Evaluated minor configuration options

Name	Value	Description
Resource Configuration Option (Hardware PDC) enabled	• YES • NO	Reconfiguration during card personalization enabled or not.
Contactless Communication	• ATQ0 Value	Defines contactless communication protocol parameters.

Name	Value	Description
Parameters	• ATQ1 Value • SAK Value • TA Value	
ROM read instructions executed from EEPROM allowed	• YES • NO	Instructions executed from EEPROM are allowed or not to read ROM contents.
ROM read instructions by Copy Machine allowed	• YES • NO	Read access by Copy Machine to ROM is allowed or not.
EEPROM read instructions by Copy Machine allowed	• YES • NO	Read access by Copy Machine to EEPROM is allowed or not.
code execution from RAM allowed	• YES • NO	Code execution from RAM allowed or not.
Activation of “Card Disable” feature allowed	• YES • NO	When the Card Disable Function is allowed, the TOE can be locked completely. Once set by the Security IC Embedded Software, execution of the Security IC Embedded Software is inhibited after the next reset.
EEPROM application content erase allowed	• YES • NO	Erase of application content of EEPROM allowed or not.
EDATASCALE specification	• EDATA size will be EDATASCALE * 16 bytes	This value determines the size of the memory area available for the extended stack pointer. Default is 10h
Inverse EEPROM Error Correction Attack Detection activated	• YES • NO	If inverse error correction is activated the detection probability of fault injections to the EEPROM can be increased.
Access to additional general purpose I/O pads TP1 and TP2 allowed in System Mode	• YES • NO	Additionally 2 general purpose I/O pads (TP1/TP2) can be accessed by the application OS.
CXRAM parity watchdog error configuration	• disabled (always off) • enabled after watchdog initialization done • enabled (always on)	Configuration of the CXRAM parity watchdog.
FXRAM parity watchdog error configuration	• disabled (always off) • enabled after	Configuration of the FXRAM parity watchdog.

Name	Value	Description
	watchdog initialization done • enabled (always on)	
Selection of reset value for UART CRC algorithm	• ISO13239/ HDLC • de facto PC/SC	Selection of CRC algorithm for ISO7816 enhanced protocol support.
Start-Up with Low CPU clock enabled	• YES • NO	Start-Up with low CPU clock to enable specific low power applications. If this option is enabled the ISO start-up timing is not met.
RunMode enabled	• YES • NO	Special start-up behavior in order to support a start-up with • RST_N pad forced to LOW or not connected and • CLK pad forced HIGH or not connected.
Allow simultaneous operation of ISO7816 and ISO14443 applications	• YES • NO	Disables the Low Frequency Sensor to allow parallel operation via contact and contactless interfaces. The Low Frequency Sensor is disabled only when the CPU is free-running or runs at an internal clock.
Chip Health mode enabled	• YES • NO	Activation of read-out of IC identification items and start of built-in self test and ident routines is enabled or not.
L_A / L_B input capacitance configuration	• 17 pF • 69 pF	Additional capacitance (2x26 pF) between LA/LB required meeting resonance frequency at ID1/2 operation.

See Section 31.1 of the “Product Data Sheet SmartMX2 family P60D012/016/024 Secure high-performance smart card controller, NXP Semiconductors” [9] for details on all minor configuration options listed in Table 4. The availability of minor configuration options partly depends on the selected major configuration option. However in general the minor configuration options can be chosen independently.

1.4.2.3 Post Delivery Configuration

Post Delivery Configuration (PDC) can be applied by the customer himself after the TOE has been delivered to that customer. These options can be used to tailor the TOE to the specific customer requirements. The Post Delivery Configuration can be changed multiple times but must be set permanently by the customer before the TOE is delivered to phase 7 of the life cycle.

The Post Delivery Configuration for the P60D024/016/012PVB is listed in Table 5.

Table 5. Post Delivery Configuration for P60D024/016/012PVB

Name	Values	Description
EEPROM Size	Steps of 2KB	This value determines the maximum size of the EEPROM in steps of 2KB. Default EEPROM size is given by the major configuration.
CXRAM Size	Steps of 128 Byte	This value determines the maximum size of the CXRAM in steps of 128 Bytes. Default CXRAM size is given by the major configuration.

Name	Values	Description
Fame2 coprocessor	Enabled or Disabled	This value determines whether the Fame2 coprocessor is enabled or disabled. Default value is enabled.
AES	Enabled or Disabled	This value determines whether the AES coprocessor is enabled or disabled. Default value is enabled.
Contactless Interface	Enabled or Disabled	This value determines whether the Contactless interface is enabled or not. Default value is enabled.

By applying Post Delivery Configuration the Security Rows content is updated for the changed configuration options and can therefore be used for identification of the TOE after applying any Post Delivery Configuration. Further details regarding Security Rows content and identification of the TOE after applying Post Delivery Configuration refer to [9].

The Post Delivery Configuration can be accessed using chip health mode functionality in combination with the ISO/IEC 7816 contact interface.

1.4.2.4 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.

- *P60D024Ypp(p)*/9B*rrff(o)* for major configuration P60D024PVB
- *P60D016Ypp(p)*/9B*rrff(o)* for major configuration P60D016PVB
- *P60D012Ypp(p)*/9B*rrff(o)* for major configuration P60D012PVB

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* and *ff*. Variables *Y* and *o* identify the activation of the firmware emulations MIFARE Plus and MIFARE DESFire which are not part of this evaluation. The number *9* is used as Fab identifier and *B* references to the silicon Version also available at major configuration naming as VB. The variables are replaced according to the rules in Table 6.

Table 6. Variable definitions for commercial type names

Variable	Definition
Y	Emulation Option Configuration for Dual Interface Types (alpha numeric), e.g. ‘M’ for MIFARE Plus.
pp(p)	Package delivery type (alpha numeric, last character optional), e.g. “A4” for MOB4 module.
rr	ROM code number, which identifies the ROM mask.
ff	FabKey number, which identifies the EEPROM content at TOE delivery.
(o)	Size of EEPROM area for firmware emulation, e.g. ‘2’ 2kByte MIFARE Plus Emulation, optional for Dual Interface Types. In case no firmware emulation is activated for a commercial type this character is left blank.

Table 7 depicts the package types, which are supported in this Security Target, and assigns these to the major configurations. The two characters in each entry of the table stand for the variable *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 7. Supported Package Types

P60D024PVB	P60D016PVB	P60D012PVB	Description
Ux	Ux	Ux	Wafer not thinner than 50 µm (The letter “x” in “Ux” stands for a capital letter or a number, which identifies the wafer type)
Xn	Xn	Xn	Module (The letter “n” in” Xn” stands for a capital letter or a number, which identifies the module type)
A4	A4		MOB4 module
A6	A6		MOB6 module
Ai	Ai		Inlay (The letter ‘i” in “Ai” stands for a capital letter, which identifies both, the inlay type and the package type inside the inlay.)

For example, commercial type name P60D024PX0/9Brrff denotes major configuration P60D024PVB in PDM1.1 dual interface smart card module. The characters ‘rr’ and ‘ff’ are individual for each customer product.

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 and in which environment 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 connections on his own.

Security during development and production is ensured for all package types listed above, for details refer to section 1.4.4.

The commercial type name identifies major configuration and package type of the TOE as well as the Security IC Embedded Software. However, the commercial type name does not itemize the minor configuration options of the TOE, which are introduced in section 1.4.2.2. Instead, minor configuration options are identified in the Order Entry Form, which is assigned to the ROM code number and the FabKey number of the commercial type name.

Minor configuration options as well as configuration options changed by means of Post Delivery Configuration are coded in the Security Rows and can be read out for identification of the TOE. Further details regarding Security Rows content and identification of the TOE after applying Post Delivery Configuration refer to [9].

1.4.3 Logical Scope of TOE

1.4.3.1 Hardware Description

The CPU of the P60D024/016/012PVB supports a 32-/24-/16-/8-bit instruction set and distinguishes five CPU modes, which are summarized in Table 8.

Table 8. CPU modes of the TOE

Super System Mode				
Boot Mode	Test Mode	Firmware Mode	System Mode	User Mode

Boot Mode, Test Mode and Firmware Mode are sub-modes of the so-called Super System Mode. These three modes are not available to the Security IC Embedded Software; they are reserved for the Security IC Dedicated Software. The Security IC Dedicated Software is composed of Security IC Dedicated Test Software and Security IC Dedicated Support Software, (Boot-ROM Software and Firmware Operating System) 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 Test Mode the TOE executes the IC Dedicated Test Software and in Firmware Mode the TOE executes the Firmware Operating System. 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 Firmware Mode.

The P60D024/016/012PVB is able to control two different logical phases. After production of the Security IC every start-up or reset completes with Test Mode and execution of the IC Dedicated Test 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.

In case the minor configuration option ‘Post Delivery Configuration’ is enabled and not finally locked by the customer, the resource configuration functionality allows the customer to enable or disable specific functionality of the hardware platform, refer to Table 5.

In case the minor configuration option ‘Chip Health/Ident Mode’ is enabled, during the boot process routines either starting built-in self tests checking the functional integrity of the TOE or sending back identification items of the TOE can be activated by the user.

System Mode and User Mode are available to the developer of the Security IC Embedded Software. System Mode has unlimited access to the hardware components available to the Security IC Embedded Software. User Mode has restricted access to the CPU, specific Special Function Registers and the memories depending on the access rights 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.

The P60D024/016/012PVB 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 addition, the P60D024/016/012PVB provides eight firmware vectors (FVEC) and 32 system call vectors (SVEC). These vectors have to be explicitly called by the Security IC Embedded Software. A jump to a firmware vector forces Firmware Mode and starts execution of the Firmware Operating System, a jump to a system call vector forces System Mode.

The Watchdog timer is intended to abort irregular program executions by a time-out mechanism and is enabled and configured by the Security IC Embedded Software.

The P60D024PVB incorporates 352 kBytes of ROM, 8.125 kBytes of RAM and 24 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 88 kBytes Test-ROM. 512 Bytes of the EEPROM are always reserved for the manufacturer area, 768 Bytes are always reserved for IC Dedicated Support Software. The Security IC Dedicated Support

Software contains functionality for programming the user EEPROM which must be called by the Security IC Embedded Software. Therefore the Security IC Dedicated Support Software has access also to the EEPROM area which is allocated to the Security IC Embedded Software, the separation between user data and NXP firmware data is guaranteed by means of a software firewall. 512 Bytes of RAM is allocated for the Firmware Operating System and the remaining part for the application. Note that the ROM size is displayed as 264 kBytes in the block diagram in Fig 1 because only 264 kBytes are available to the Security IC Embedded Software.

In Test Mode the CPU has unrestricted access to all memories. In Boot Mode and Firmware Mode access is limited to the Test-ROM, the manufacturer area of the EEPROM and its configured part of 768 Bytes as well as the configured part of 512 Bytes RAM for the Firmware Operating System. 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 5.0 kBytes general purpose RAM (CXRAM) and 2.625 kBytes FXRAM (associated to the Fame2 coprocessor). Both parts are accessible to the CPU, but the Fame2 coprocessor can only access the FXRAM. The Fame2 coprocessor can access the FXRAM without control of access rights by the Memory Management Unit. Since the Memory Management Unit does not control accesses of the Fame2 coprocessor, software which has access to the Fame2 coprocessor implicitly has access to the FXRAM.

The Triple-DES coprocessor supports single DES and Triple-DES operations. Only Triple-DES is in the scope of this evaluation, in 2-key or 3-key operation with two/three 56-bit keys (112-/168-bit). The AES coprocessor supports AES operation with three different key lengths of 128, 192 or 256 bit. The Fame2 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. The CRC coprocessor provides CRC generation polynomial CRC-16 and CRC-32. The copy machine supports a mechanism to transfer data between specific Special Function Registers as well as memories without interaction of the CPU.

The P60D024PVB operates with a single external power supply of 1.8 V, 3 V or 5 V nominal. Alternatively the P60D024PVB can be supplied via the RF interface by inductive coupling. The maximum external clock frequency used for synchronization of the ISO/IEC 7816 communication is 10 MHz nominal, the CPU and all co-processors are supplied exclusively with an internally generated clock signal which frequency can be selected by the Security IC Embedded Software. The P60D024PVB 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. A memory encryption is added to the memories RAM, ROM and EEPROM. EEPROM double read function is included in this memory to check data consistency during EEPROM read. Chip shielding is added in form of active and passive shield over logic and memories. Sensors in form of light, voltage, temperature and frequency sensors are distributed over the chip area. The security functionality of the IC hardware platform is mainly 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.

1.4.3.2 Software Description

Operating system and applications of a Security IC are developed by the customers and included under the heading 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 IC hardware platform.

The Security IC Dedicated Test Software, is stored to the Test-ROM and used by the manufacturer of the Security IC during production test. The test functionality is disabled before the TOE is delivered (operational use of the Security IC) by disabling the Test Mode of the CPU in hardware. The Security IC Dedicated Test Software is developed by NXP and embedded in the Test-ROM. The Security IC Dedicated Test Software includes the test operating system, test routines for the various blocks of the circuitry, control flags for the status of the EEPROM's manufacturer area and shutdown functions to ensure that security relevant test routines can not be executed illegally after phase 3.

The Security IC 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.
- The Firmware Operating System used as interface for firmware provided by NXP. This comprises the control of hardware related functionality such as access to the manufacturer area of the EEPROM, programming of user data into the EEPROM or the resource configuration functionality.

The execution of the Firmware Operating System is separated by security mechanism implemented in the hardware including the firewall separation of the Firmware Mode controlling the access to memories and Special Function Register as configured in hardware or by the Security IC Embedded Software.

The TOE is always delivered with a Firmware Operating System. The related functionality is part of the hardware platform evaluation. The Firmware Operating System of the TOE is limited to the control of hardware related functionality and the resource configuration functionality.

1.4.3.3 Documentation

The data sheet "Product Data Sheet SmartMX2 family P60D012/016/024 Secure high-performance smart card controller, NXP Semiconductors" [9] contains a functional description and guidelines for the use of the security functionality, as needed to develop Security IC Embedded Software. The documents "Product data sheet addendum: SmartMX2 family Chip Health Mode (CHM), NXP Semiconductors" [17] and "Product data sheet addendum: SmartMX2 family, Post Delivery Configuration (PDC), NXP Semiconductors" [16] contain description and guidelines in addition to [9] for Chip Health Mode and Post Delivery Configuration. The instruction set of the CPU is described in "Instruction Set for the SmartMX2 family, Secure smart card controller" [10]. The manual "NXP Secure Smart Card Controller P60D024/016/012 VB Guidance and Operation Manual" [11] describes aspects of the program interface and the use of programming techniques to improve the security. The wafer and delivery specification "SmartMX2 P60D012/016/024 VB Wafer and delivery specification, NXP Semiconductors, Business Unit Identification" [12] describes physical identification of the TOE and the secure delivery process. The whole documentation shall be used by the developer to develop the Security IC Embedded Software.

1.4.4 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. Phase 4 the IC Packaging is also part of the evaluation. The Security IC is delivered at the end of phase 3 or phase 4 in the life-cycle. 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 “Chip Health/Ident Mode” and the FabKey feature. For further details on these features please refer to the data sheet [9] and the guidance and operation manual [11].

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 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, including the ROM Code, and the remaining mask set.

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 TOE. 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 four different forms, which are

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

The availability of major configuration options of the TOE in package types is detailed in section 1.4.2.4.

1.4.5 TOE Intended Usage

The end-consumer environment of the TOE is phase 7 of the Security IC product life-cycle as defined in the PP [6]. In this phase the Security IC product is in usage by the end-consumer. Its method of use now depends on the Security IC Embedded Software. The Security ICs including the P60D024/016/012PVB can be 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 [21] and for contactless applications according to ISO/IEC 14443 [23]. 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.

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 behaviour of the Security IC 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 from TOE Delivery to 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 can not be disabled by the Security IC Embedded Software in the following phases.

1.4.6 Interface of the TOE

The electrical interface of the P60D024/016/012PVB are the pads to connect the lines power supply, ground, reset input, clock input, serial communication pad I/O1 and depending on a minor configuration option TP1 and TP2, as well as two pads (called LA and LB) for the antenna of the RF interface. Communication with the TOE can be established via the contact interface through the ISO/IEC 7816 UART or direct usage of the I/O ports. Contactless communication is done via the contactless interface unit (CIU) compatible to ISO/IEC 14443.

The logical interface of the TOE depends on the CPU mode and the associated software.

- In Boot Mode the Boot-ROM Software is executed. Only in case the minor configuration option “Chip Health/Ident Mode” is enabled, starting of built-in self test routines and read-out of TOE identification items is supported. If this minor configuration option is disabled the Boot-ROM Software provides no interface. In this case there is no possibility to interact with this software.
- In Test Mode (used before TOE delivery) the logical interface visible on the electrical interface is defined by the Security IC Dedicated Test Software. This Security IC Dedicated Test Software comprises the test operating system and the package of test function calls.
- In Firmware Mode the Firmware Operating System is executed by the CPU. The Firmware Mode is always requested by the Security IC Embedded Software.
- In System Mode and User Mode (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 that is visible on the electrical interface after TOE Delivery is based on the Security IC Embedded Software developed by the software developer. The identification and authentication of the user in System Mode or User Mode must be controlled by the Security IC Embedded Software.

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, for which the attacker manipulates the chip surface.

Note: An external voltage and timing supply as well as a logical interface are necessary for the operation of the TOE. Beyond the physical behaviour the logical interface is defined by the Security IC Embedded Software.

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 and the TOE claims to be conformant to version 3.1 of Common Criteria for Information Technology Security Evaluation according to

- “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 for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 3, July 2009, CCMB-2009-07-002” [2]
- “Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, 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 and the TOE claims to be CC Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in Chapter 5.

2.2 Package claim

This Security Target claims conformance to the assurance package EAL6 augmented. The augmentation to EAL6 is ALC_FLR.1. In addition, the assurance package of this 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 Protection Profile” [6] to which this Security Target claims conformance (for details refer to section 2.3) requires assurance level EAL4 augmented. The changes, which are needed for EAL6, are described in the relevant sections of this Security Target.

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 strict 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 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 the PP [6], this additional functionality is added using the policy “P.Add-Components” (see Section 3.3 of this Security Target for details).

2.4 Conformance Claim Rationale

According to Section 2.3, this Security Target claims strict conformance to the PP “Security IC Protection Profile [6].

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

All sections of this Security Target, in which 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 (EAL6+) is shown in section 6.2 to include respectively exceed the requirements claimed by the PP [6] (EAL4+).

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

3. Security Problem Definition

This Security Target claims conformance to the PP “Security IC Protection Profile” [6]. Assets, threats, assumptions and organisational security policies are taken from the PP [6]. This chapter lists these assets, threats, assumptions and organisational 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 strict conformance to the PP “Security IC Protection Profile” [6] the assets defined in section 3.1 of [6] are applied here. These assets are cited below.

The assets related to standard functionality are:

- integrity and confidentiality of User Data stored and in operation,
- integrity and confidentiality of Security IC Embedded Software, stored and in operation,
- correct operation of the security services and restricted hardware resources provided by the TOE for the Security IC Embedded Software.

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

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

Note that the keys for the cryptographic calculations using cryptographic coprocessors are seen as User Data.

3.2 Threats

Since this Security Target claims strict conformance to the PP “Security IC Protection Profile” [6] the threats defined in section 3.2 of [6] are valid for this Security Target. The threats defined in the PP [6] are listed below in Table 9.

Table 9. 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
T.Leak-Forced	Forced Information Leakage
T.Abuse-Func	Abuse of Functionality
T.RND	Deficiency of Random Numbers

Considering Application Note 5 in [6] the TOE provides additional functionality to protect against threats that may occur if the hardware platform is used for multiple applications. The TOE provides access control to the memories and to hardware resources providing security services for the software.

The Security IC Embedded Software controls all User Data stored by the TOE. If multiple applications are running on the TOE the User Data may belong to different applications. The access to User Data from application A by the application B contradicts the separation between the different applications and is considered as threat. The User Data is stored in the memory and processed by the hardware resources.

The TOE shall avert the threat “Unauthorised Memory or Hardware Access (T.Unauthorised-Access)” as specified below.

T.Unauthorised-Access Unauthorised Memory or Hardware Access

Adverse action: An attacker may try to read, modify or execute code or data stored in restricted memory areas. And or an attacker may try to access or operate hardware resources that are restricted by executing code that accidentally or deliberately accesses these restricted hardware resources.

Any code or data executed in Boot Mode, Firmware Mode, System Mode or User Mode may accidentally or deliberately access User Data or code of another application stored on the TOE. Or any code or data executed in Boot Mode, Firmware Mode, System Mode or User Mode may accidentally or deliberately access hardware resources that are restricted or reserved for other CPU modes.

Threat agent: having high attack potential and access to the TOE

Asset: execution of code or data belonging to the Security IC Dedicated Support Software as well as belonging to Security IC Embedded Software.

Access restrictions for the memories and hardware resources accessible by the Security IC Embedded Software must be defined and implemented by the security policy of the Security IC Embedded Software based on the specific application context.

Table 10. Additional threats averted by the TOE

Name	Title
T.Unauthorised-Access	Unauthorised Memory or Hardware Access

3.3 Organisational Security Policies

Since this Security Target claims strict conformance to the PP “Security IC 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] there is one additional policy 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 IC Developer/Manufacturer therefore applies the policy "Additional Specific Security Components (P.Add-Components)" as specified below.

P.Add-Components Additional Specific Security Components

The TOE shall provide the following additional security functionality to the Security IC Embedded Software:

- Triple-DES encryption and decryption
- AES encryption and decryption
- Integrity support of data stored in EEPROM
- Post Delivery Configuration: reconfiguration of customer selectable options as listed in Table 5 (for the P60D024PVB/P60D016PVB/P60D012PVB).

3.4 Assumptions

Since this Security Target claims strict conformance to the PP "Security IC 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 11. Assumptions defined in the PP [6]

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

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

A.Check-Init Check of initialisation data by the Security IC Embedded Software

The Security IC Embedded Software must provide a function to check initialisation data. The data is defined by the customer and 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 specialised 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.

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 Protection Profile” [6].

Table 12. Security objectives defined in the PP [6]

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

Regarding Application Notes 9 and 10 in [6] the following additional security objectives are defined based on additional functionality provided by the TOE as specified below.

O.INTEGRITY_CHK — Integrity control of transferred data

The TOE shall provide a CRC coprocessor that supports the integrity protection of user data and TSF data transferred between different parts of the TOE. This comprises data transfer between the memories or between a memory and a hardware component of the TOE.

Note: The integrity control provided by the TOE shall only be active if explicitly configured by the Security IC Embedded Software.

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. 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. 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.CUST_RECONFIG	Post Delivery Configuration The TOE shall provide the customer with the functionality to reconfigure parts of the TOE properties as specified for the Post Delivery Configuration listed in Table 5 (for the P60D024PVB/P60D016PVB/P60D012PVB).
O.EEPROM_INTEGRITY	Integrity support of data stored in EEPROM The TOE shall provide a retrimming of the EEPROM to support the integrity of the data stored in the EEPROM.
O.FM_FW	Firmware Mode Firewall The TOE shall provide separation between the NXP Firmware (i.e. NXP firmware functionality as part of the Security IC Dedicated Support Software) as part of the Security 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, Firmware 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 specialised hardware components of the TOE shall be restricted to code running in System Mode.

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 Protection Profile” [6] defines security objectives for the Security IC Embedded Software development environment which are listed in the table below.

Table 13. Security objectives for the Security IC Embedded Software development environment, taken from the PP [6]

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 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 realised 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 Protection Profile" [6].

Table 14. 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

Check of initialisation data

The TOE provides specific functionality that requires the TOE Manufacturer to implement measures for the unique identification of the TOE. Therefore, OE.Check-Init is defined to allow a TOE specific implementation (refer also to A.Check-Init).

OE.Check-Init	Check of initialisation 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-personalisation data. This shall include at least the FabKey Data that is agreed between the customer and the TOE Manufacturer.

4.4 Security Objectives Rationale

Section 4.4 in the PP “Security IC Protection Profile” [6] provides a rationale how the assumptions, threats, and organisational security policies are addressed by the objectives that are specified in the PP [6]. Table 15 reproduces the table in section 4.4 of [6].

Table 15. 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	Phases 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	

Table 16 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, threat and organisational security policy.

Table 16. Additional Security Objectives versus Assumptions, Threats or Policies

Assumption/Threat/Policy	Security objective	Notes
T.Unauthorised-Access	O.FM_FW O.MEM_ACCESS O.SFR_ACCESS	
T.Malfunction	O.INTEGRITY_CHK	Based on the PP the security objective O.Malfunction is already mapped to this threat.

Assumption/Threat/Policy	Security objective	Notes
P.Add-Components	O.HW_DES3 O.HW_AES O.CUST_RECONFIG O.EEPROM_INTEGRITY	
A.Key-Function	OE.Plat-Appl OE.Resp-Appl	Phase 1
A.Check-Init	OE.Check-Init	Phase 1 and Phases 4 - 6

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

The justification related to the threat “Unauthorised Memory or Hardware Access (T.Unauthorised-Access)” is as follows:

According to O.FM_FW, O.MEM_ACCESS and O.SFR_ACCESS the TOE must enforce the partitioning of memory areas in Firmware Mode, System Mode and User Mode and enforce the segmentation of the memory areas in User Mode so that access of software to memory areas is controlled. Any restrictions have to be defined by the Security IC Embedded Software. Thereby security violations caused by accidental or deliberate access to restricted data (which may include code) can be prevented (refer to T.Unauthorised-Access). The threat T.Unauthorised-Access is therefore covered by the objective.

The clarification of “Usage of Hardware Platform (OE.Plat-Appl)” makes clear that it is up to the Security IC Embedded Software to implement the memory management scheme by appropriately administrating the TSF. This is also expressed both in T.Unauthorised-Access and O.FM_FW, O.MEM_ACCESS and O.SFR_ACCESS. The TOE shall provide access control functions to be used by the Security IC Embedded Software. This is further emphasised by the clarification of “Treatment of User Data (OE.Resp-Appl)” which reminds that the Security IC Embedded Software must not undermine the restrictions of the hardware platform. Therefore, the clarifications contribute to the coverage of the threat T.Unauthorised-Access.

The justification related to the security objectives O.INTEGRITY_CHK is as follows: Since the objective provides the functionality to check the integrity of user data and TSF data during the transfer between different parts of the TOE the objective implements specific security functionality to detect the manipulation of user data or TSF data. Thereby the threat T.Malfunction is removed. Therefore the threat is countered if the objective holds.

The justification related to the security objectives O.HW_DES3, O.HW_AES, O.CUST_RECONFIG and O.EEPROM_INTEGRITY is as follows: Since these objectives require the TOE to implement exactly the same specific security functionality as required by P.Add-Components, the organisational security policy is covered by the objectives.

Nevertheless the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced define how to implement the specific security functionality required by P.Add-Components. These security objectives are also valid for the additional specific security functionality since they must avert the related threats also for the components added related to the policy.

The requirements for a multi-application platform necessitate the separation of users. Therefore it is volitional that most of the security functionality can not be influenced or used in User Mode.

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.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.

The justification related to the assumption "Check of initialisation data by the Security IC Embedded Software (A.Check-Init)" is as follows:

Since OE.Check-Init requires the Security IC Embedded Software developer to implement a function assumed in A.Check-Init, the assumption is covered by the objective.

The justification of the additional policy and the additional assumptions show that they do not contradict to the rationale already given in the PP [6] for the assumptions, policy and threats defined there.

5. Extended Components Definition

This Security Target does not define extended components.

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

The extended component definition used for Random Number Generator has been taken from [8].

6. Security Requirements

This part of the Security Target defines the detailed security requirements that shall be satisfied by the TOE. The statement of TOE security requirements shall define the functional and assurance security requirements that the TOE needs to satisfy in order to meet the security objectives for the TOE. This chapter consists of the sections “Security Functional Requirements”, “Security Assurance Requirements” and “Security Requirements Rationale”.

The CC allows several operations to be performed on security requirements (on the component level); refinement, selection, assignment, and iteration are defined in paragraph 8.1 of Part 1 of the CC [1]. These operations are used in the PP [6] and in this Security Target, respectively.

The **refinement** operation is used to add details to requirements, and, thus, further restricts a requirement. Refinements of security requirements are denoted in such a way that added words are in bold text and changed words are crossed out.

The **selection** operation is used to select one or more options provided by the PP [6] or CC in stating a requirement. Selections having been made are denoted as italic text.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a password. Assignments having been made are denoted by showing as italic text.

The **iteration** operation is used when a component is repeated with varying operations. It is denoted by showing brackets “[*iteration indicator*]” and the *iteration indicator* within the brackets.

For the sake of a better readability, the iteration operation may also be applied to some single components (being not repeated) in order to indicate belonging of such SFRs to same functional cluster. In such a case, the iteration operation is applied to only one single component.

Whenever an element in the PP [6] contains an operation that the PP author left uncompleted, the ST author has to complete that operation.

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 Protection Profile” [6] and Security Target.

6.1.1 SFRs of the Protection Profile

Table 17 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 17. 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

SFR	Title	Defined in
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

The definition of the SFRs FDP_ITT.1 and FPT_ITT.1 is repeated in this Security Target because the selection in each SFR is extended. Based on the Data Processing Policy defined in PP [6] the SFRs FDP_ITT.1 and FPT_ITT.1 include the additional requirement to prevent modification of user data and TSF data. The Refinement for the physically separated parts of the TOE is still valid for both SFRs.

The TOE shall meet the requirement “Basic internal transfer protection (FDP_ITT.1)” as specified below.

FDP_ITT.1[HW]	Basic internal transfer protection
Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_ITT.1.1[HW]	The TSF shall enforce the *Data Processing Policy*[2] to prevent the *disclosure and modification*[3] of user data when it is transmitted between physically-separated parts of the TOE.
Refinement:	**The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic or CRC co-processor) are seen as physically-separated parts of the TOE.**

The TOE shall meet the requirement “Basic internal TSF data transfer protection (FPT_ITT.1)” as specified below.

FPT_ITT.1[HW]	Basic internal TSF data transfer protection
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_ITT.1.1[HW]	The TSF shall protect TSF data from *disclosure and modification*[4] when it is transmitted between separate parts of the TOE.
Refinement:	**The different memories, the CPU and other functional units of the TOE (e.g. a cryptographic or CRC co-processor) are seen as separated parts of the TOE.**

The operations for all other SFR except for the SFR FAU_SAS.1 and FCS_RNG.1 are already performed in the PP [6]. They are not changed compared to the Protection Profile. The open assignments and selections for FAU_SAS.1 and FCS_RNG.1 are included in the following.

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

3 [selection: *disclosure, modification, loss of use*]

4 [selection: *disclosure, modification*]

For the SFR FAU_SAS.1 the PP [6] leaves the assignment operation open for the non-volatile memory type in which initialisation data, pre-personalisation 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[HW]	Audit storage
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FAU_SAS.1.1[HW]	The TSF shall provide *the test process before TOE Delivery* [5] with the capability to store *the Initialisation Data and/or Pre-personalisation Data and/or supplements of the Security IC Embedded Software* [6] in the *EEPROM* [7].

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 operations defined in chapter 3 of [8] and the open operations of the partially filled in operations in the statement of the security requirements in section 4.4 of [8] for better readability. Note that the selection operation for the RNG type has already been filled in by the PP [6].

FCS_RNG.1[HW]	Random number generation (Class PTG.2)
Hierarchical to:	No other components.
Note:	The definition of the Security Functional Requirement FCS_RNG.1 has been taken from [8].
Note:	The functional requirement FCS_RNG.1[HW] is a refinement of FCS_RNG.1 defined in PP [6] according to [8].
FCS_RNG.1.1[HW]	The TSF shall provide a *physical* [8] random number generator that implements:
(PTG.2.1)	*A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure is detected, no random numbers will be output.*
(PTG.2.2)	*If a total failure of the entropy source occurs while the RNG is being operated, the RNG prevents the output of any internal random number that depends on some raw random numbers*

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

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

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

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

that have been generated after the total failure of the entropy source[9].

(PTG.2.3) The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG has started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test has finished successfully or when a defect has been detected.

(PTG.2.4) The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.

(PTG.2.5) The online test procedure checks the quality of the raw random number sequence. It is triggered at regular intervals or continuously[10]. The online test is suitable for detecting non-tolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time[11].

Note: The TOE provides the two options where the Embedded Software can choose one

FCS_RNG.1.2[HW] The TSF shall provide *octets of bits*[12] that meet:

(PTG.2.6) Test procedure A[13] does not distinguish the internal random numbers from output sequences of an ideal RNG.

(PTG.2.7) The average Shannon entropy per internal random bit exceeds 0.997.

Note: The Shannon entropy 0.997 per internal random bit compares to 7.976 per octet

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.

9 [selection: *prevents the output of any internal random number that depends on some raw random numbers that have been generated after the total failure of the entropy source, generates the internal random numbers with a post-processing algorithm of class DRG.2 as long as its internal state entropy guarantees the claimed output entropy]*

10 *[*selection*: externally, at regular intervals, continuously, applied upon specified internal events]*

11 [assignment: *list of security capabilities*]

12 [selection: *bits, octets of bits, numbers [assignment: format of the numbers]]*

13 [assignment: *additional standard test suites*] Note: according §295 in [8] the assignment may be empty

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”, [7].

Dependencies: No dependencies.

By this, all assignment/selection operations are performed. This Security Target does not perform any other/further operations than stated in [8].

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

As required by Application Note 14 of the PP [6] the secure state is described in Section 7.2.2 in the rationale for SF.OPC.

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

As required by Application Note 18 of the PP [6] the automatic response of the TOE is described in Section 7.2.2 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)” as specified below.

FCS_COP.1[HW_DES] Cryptographic operation

Hierarchical to: No other components.

FCS_COP.1.1[HW_DES] The TSF shall perform *encryption and decryption*[14] in accordance with a specified cryptographic algorithm *Triple Data Encryption Algorithm (TDEA)*[15] and cryptographic key sizes *of 112 or 168 bit*[16] that meet the following *list of standards*[17]:

FIPS PUB 46-3 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION DATA ENCRYPTION STANDARD (DES) Reaffirmed 1999 October 25, keying options 1 and 2 [18].

Dependencies: [FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or

[14] [assignment: *list of cryptographic operations*]

[15] [assignment: *cryptographic algorithm*]

[16] [assignment: cryptographic key sizes]

[17] [assignment: list of standards]

	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.
Note:	The security functionality is resistant against side channel analysis and similar techniques. To fend off attackers with high attack potential a security level of at least 80 Bits must be used.

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

FCS_COP.1[HW_AES]	**Cryptographic operation**
Hierarchical to:	No other components.
FCS_COP.1.1[HW_AES]	The TSF shall perform *encryption and decryption*[18] in accordance with a specified cryptographic algorithm *Advanced Encryption Standard (AES) algorithm*[19] and cryptographic key sizes *of 128, 192 or 256 bit*[20] that meet the following *list of standards*[21]: *FIPS PUB 197 FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, ADVANCED ENCRYPTION STANDARD (AES), National Institute of Standards and Technology, 2001 November 26 [19].*
Dependencies:	[FDP_ITC.1 Import of user data without security attributes or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation], FCS_CKM.4 Cryptographic key destruction.

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

FDP_SDI.2[HW]	**Stored data integrity monitoring and action**
Hierarchical to:	FDP_SDI.1 Stored data integrity monitoring
FDP_SDI.2.1[HW]	The TSF shall monitor user data stored in containers controlled by the TSF for *integrity violations due to ageing*[22] on all objects, based on the following attributes: *User data including code stored in the EEPROM*[23].
FDP_SDI.2.2[HW]	Upon detection of a data integrity error, the TSF shall *adjust the EEPROM write operation*[24].
Dependencies:	No dependencies.

18 [assignment: list of cryptographic operations]

19 [assignment: cryptographic algorithm]

20 [assignment: cryptographic key sizes]

21 [assignment: list of standards]

22 [assignment: integrity errors]

23 [assignment: user data attributes]

24 [assignment: action to be taken]

Refinement: **Each EEPROM memory block is considered as one container and the adjustment is done for one complete EEPROM memory block.**

6.1.3 Additional SFRs regarding access control

Access Control Policy

The hardware shall provide different CPU modes to Security IC Dedicated Support Software and Security IC Embedded Software for separating the code and data of these two domains. The separation shall be supported by the partitioning of memories. Management of access to code and data as well as access to hardware resources shall be assigned to dedicated CPU modes. The hardware shall enforce separation between different applications (i.e. parts of the Security IC Embedded Software) running on the TOE. The TOE shall support this based on the CPU modes and the segmentation of the memories. The TOE shall support secure operation on Special Function Register depending on register functionality or on the CPU mode. In addition an application shall not be able to access hardware components unless permission is granted explicitly. The hardware shall provide direct memory access for the Security IC Embedded Software without CPU interactions realized by a copy machine. The copy machine shall support different CPU modes and the segmentation of the memories.

The Security Function Policy (SFP) 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 Security IC Dedicated Test Software, executed as instructions by the CPU
- The Boot-ROM Software as part of the Security IC Dedicated Support Software, executed as instructions by the CPU
- The Firmware Operating System as part of the Security IC Dedicated Support Software including the resource configuration firmware executed as instructions by the CPU and stored data integrity monitoring for EEPROM write accesses of the Security IC Embedded Software
- The Firmware Firewall configured by the Security IC Dedicated Support Software for restricted access of the Firmware Operating System to the hardware related Special Function Registers and separation between Security IC Dedicated Support Software and Security IC Embedded Software
- The copy machine configured by the Security IC Embedded Software for direct memory access enforcing separation between different CPU modes and the segmentation of memories
- The Fame2 coprocessor configured by the Security IC Embedded Software for implementation of asymmetric cryptographic algorithms and direct memory access to the FXRAM for accessing operands and storing resulting data.

The objects are

- the memories consisting of
  - ROM, which is partitioned into Test-ROM and Application-ROM

- EEPROM, which is partitioned into two parts. To simplify referencing, the part reserved for the Firmware Operating System is called Firmware-EEPROM, the other part Application-EEPROM.
- RAM, which is partitioned into two parts. To simplify referencing, the part reserved for the Firmware Operating System is called Firmware-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.

- The virtual memory locations within the three memories that are used by the CPU and are mapped to physical addresses by the Memory Management Unit.
- 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 Firmware firewall. These Special Function Registers allow modifying the Firmware firewall regarding data exchange and Special Function Register access control.
  - Special Function Registers used by the Firmware Operating System including the resource configuration firmware. The Firmware Operating System 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 utilise 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.
    - Special Function Registers related to general CPU functionality implemented separately for System and User Mode. This group contains CPU watch exception register for System and User Mode.
- The Firmware Firewall configured during bootflow that separates memories of Firmware Operating System from Security IC Embedded Software

The memory operations are

- read data from the memory,
- write data into the memory and
- execute data stored 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 that are sequentially active based on the configuration of Special Function Registers defining whether the instruction is executed in Boot Mode, Test Mode, Firmware 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.
- 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 of 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 MMU_FWCTRLL, MMU_FWCTRLH, MMU_MXBASL, MMU_MXBASH, MMU_MXSZL and MMU_MXSZH belonging to the group Special Function Registers related to hardware components that define the access rights to the Special Function Registers related to hardware components for code executed in Firmware Mode and the RAM area used for data exchange between Security IC Dedicated Support Software (Firmware Operating System including resource configuration firmware) 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 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[MEM]	The TSF shall enforce the *Access Control Policy*[25] on *all code running on the TOE, all memories and all memory operations*[26].
Dependencies:	FDP_ACF.1 Security attribute based access control
Application Note:	The 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.

[25] [assignment: access control SFP]

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

FDP_ACC.1.1[SFR]	The TSF shall enforce the *Access Control Policy*[27] on *all code running on the TOE, all Special Function Registers, and all Special Function Register operations*[28].
Dependencies:	FDP_ACF.1 Security attribute based access control
Application Note:	The 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 Firmware Mode the access rights to the Special Function Registers related to hardware components are provided by the MMU Segment Table and the Special Function Registers to configure the Firmware firewall. A denied read or write access triggers an exception. 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[MEM]	The TSF shall enforce the *Access Control Policy*[29] 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*[30].
FDP_ACF.1.2[MEM]	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 Firmware-EEPROM* • *has read and write access to all data in the Firmware-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*

[27] [assignment: access control SFP]

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

[29] [assignment: access control SFP]

[30] [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 and write access to all data in the whole RAM*

Code executed in the Firmware Mode

- *has read and execute access to its own code/data in the Firmware-ROM,*
- *has read and write access to all code/data in the whole EEPROM for data integrity control during EEPROM write operations and read, write and execute access to the Firmware EEPROM for other purpose controlled by the Firmware Firewall has read and write access to all data in the Firmware-RAM.*

Code executed in the System Mode

- *has read and execute access to all code/data in the Application-ROM,*
- *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.* [31]

FDP_ACF.1.3[MEM] The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *Code running in Firmware Mode has access to the Application-RAM defined by the Special Function Register MMU_MXBASL, MMU_MXBASH, MMU_MXSZL and MMU_MXSZH. Code running in Boot Mode or Firmware Mode has read access to the Security Rows stored in the Application-EEPROM. Code running in Firmware Mode when called from System Mode has read and write access to the Application-EEPROM for data integrity control reasons during EEPROM write operations. The Fame2 coprocessor has read/write access to the FXRAM*[32]*.*

FDP_ACF.1.4[MEM] The TSF shall explicitly deny access of subjects to objects based on the following additional rules*: if configured code executed in EEPROM cannot read ROM, if configured the*

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

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

copy machine cannot read ROM, if configured the copy machine cannot read EEPROM[33].

Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialisation
FDP_ACF.1[SFR]	**Security attribute based access control**
Hierarchical to:	No other components.
FDP_ACF.1.1[SFR]	The TSF shall enforce the *Access Control Policy*[34] to objects based on the following: *all subjects and objects and the attributes CPU mode, the MMU Segment Table and the Special Function Registers MMU_FWCTRLL and MMU_FWCTRLH*[35]."
FDP_ACF.1.2[SFR]	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 except Special Function Registers related to testing, Special Function Registers to configure the MMU segmentation and Special Function Registers related to general CPU functionality implemented separately for System and User Mode.*
- *The code executed in Test Mode is allowed to access all Special Function Register groups except Special Function Registers to configure the MMU segmentation and Special Function Registers related to general CPU functionality implemented separately for System and User Mode.*
- *The code executed in Firmware Mode is allowed to read Special Function Registers to configure the Firmware firewall and to read/write Special Function Registers used by the Firmware Operating System and the resource configuration firmware. Access to Special Function Registers related to hardware components is based on the access rights determined by the Special Function Registers MMU_FWCTRLL and MMU_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 Firmware 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*

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

[34] [assignment: access control SFP]

[35] [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]

Memory Segment in the MMU Segment Table from which the code is actually executed[36]*.*

Application Note: Copy Machine continues operation in the CPU mode in which it has been started independent of any CPU mode changes initiated by the Security IC Embedded Software during copy machine operation.

FDP_ACF.1.3[SFR] The TSF shall explicitly authorise 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, except those implemented separately for System and User Mode, is allowed. In System and User Mode access to the Special Function Registers related to general CPU functionality implemented separately for System and User Mode is allowed. The Special Function Register CPU_CSR belonging to group Special Function Registers related to system management is additionally readable in Firmware Mode and User Mode. The Special Function Register CFG_CLKSEL of the group Special Function Registers related to hardware components can be read in the Firmware Mode regardless of the Firmware firewall settings given by MMU_FWCTRLL and MMU_FWCTRLH*[37]*.*

FDP_ACF.1.4[SFR] The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *Access to Special Function Registers to configure the MMU segmentation is denied in all CPU modes except System Mode. Access to Special Function Registers related to general CPU functionality implemented separately for System and User Mode is denied in Boot, Test and Firmware Mode. The Special Function Register MMU_RPT2 of the group Special Function Registers related to system management is not readable. The Special Function Register RNG_RNR of the group Special Function Registers related to hardware components is read-only. The Special Function Registers SBC_KEY used as key registers for AES and DES coprocessors of the group Special Function Registers related to hardware components are not readable*[38]*.*

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

Implications of the 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 and used to configure and test the TOE.

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

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

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

- Code executed in Firmware 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 Firmware Operating System and the Security IC Embedded Software. Furthermore, the exchange area in RAM is fully controlled by code running in System Mode. The EEPROM data integrity function executed in Firmware Mode has access to the whole EEPROM area to guarantee data integrity for EEPROM write operations. Other Firmware functions have only access to a separated dedicated area in the EEPROM. Separation is realized by means of a Firmware Firewall.
- 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 initialisation (FMT_MSA.3)” as specified below.

FMT_MSA.3[MEM]	**Static attribute initialisation**
Hierarchical to:	No other components.
FMT_MSA.3.1[MEM]	The TSF shall enforce the *Access Control Policy* [39] to provide *restrictive* [40] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2[MEM]	The TSF shall allow *no subject* [41] 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 can not be configured at all. The TOE does not provide objects or information that can be created, since it provides access to memory areas. The definition of objects that are stored in the TOE’s memory is subject to the Security IC Embedded Software.

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

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

[41] [assignment: the authorised identified roles]

FMT_MSA.3[SFR]	**Static attribute initialisation**
Hierarchical to:	No other components.
FMT_MSA.3.1[SFR]	The TSF shall enforce the *Access Control Policy* [42] to provide *restrictive* [43] default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2[SFR]	The TSF shall allow *no subject* [44] 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[MEM]	The TSF shall enforce the *Access Control Policy* [45] to restrict the ability to *modify* [46] the security attributes *Special Function Registers to configure the MMU segmentation* [47] to *code executed in the System Mode* [48].
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.
FMT_MSA.1[SFR]	**Management of security attributes**
Hierarchical to:	No other components.

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

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

[44] [assignment: the authorised identified roles]

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

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

[47] [assignment: list of security attributes]

[48] [assignment: the authorised identified roles]

FMT_MSA.1.1[SFR]	The TSF shall enforce the *Access Control Policy* [49] to restrict the ability to *modify* [50] the security attributes *defined in Special Function Registers* [51] to *code executed in a CPU mode which has write access to the respective Special Function Registers* [52].
Dependencies:	[FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions

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

FMT_SMF.1[HW]	**Specification of Management Functions**
Hierarchical to:	No other components.
FMT_SMF.1.1[HW]	The TSF shall be capable of performing the following management functions: *Change of the CPU mode by calling a system call vector (SVEC) or firmware vector (FVEC) 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 CPU_CSR Special Function Register and* *modification of the Special Function Registers containing security attributes, and* *modification of the MMU Segment Table, and* *temporary disabling and enabling of the security functionality EEPROM Size, CXRAM Size, AES coprocessor, Fame2 coprocessor and* *permanent disabling and enabling of the security functionality EEPROM Size, CXRAM Size, AES coprocessor, Fame2 coprocessor*[53].
Dependencies:	No dependencies
Application Note:	The iteration of FMT_MSA.1 with the dependency to FMT_SMF.1[HW] may imply a separation of the Specification of Management Functions. Iteration of FMT_SMF.1[HW] is not needed because all management functions rely on the same features implemented in the hardware.

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

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

[51] [assignment: list of security attributes]

[52] [assignment: the authorised identified roles]

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

Note that the access control policy defined above also depends on the major configuration of the TOE, refer to section 1.4.2.

The following table provides an overview of the differences in the “P60Dxxx”-configuration. These must be considered for the following access control policy and the related Security Functional Requirements.

Table 18. Differences between TOE configurations with regard to the Access Control Policy

	TOE major configurations P60D024PVB, P60D016PVB, P60D012PVB	**Remark**
Access to 512 byte Security rows and manufacturer area	Boot Mode, Test Mode, System Mode, User Mode and Firmware Mode	described in detail below, function in general not influenced by the configuration
Access to 768 byte Firmware area in EEPROM	Boot Mode, Test Mode and Firmware Mode	described in detail below, function in general not influenced by the configuration
Access to RAM	Boot Mode, Test Mode, System Mode, User Mode and Firmware Mode	described in detail below, function in general not influenced by the configuration
Firmware Mode firewall for the RAM access	configuration of the Firmware Mode firewall supports the separation between Security IC Dedicated Support Software and Security IC Embedded Software and restricts the access of the Firmware operating System to 512 byte firmware image in RAM and shared data defined by IC embedded Software in System and User Mode	refer to the description of the Firmware Mode firewall below, function in general not influenced by the configuration
Firmware Mode firewall for the Special Function Register access	the configuration of the Firmware Mode firewall restricts the access of the Firmware Operating System to the hardware related Special Function Registers, Firmware functionality for data integrity control and resource reconfiguration is available.	refer to the description of the Firmware Mode firewall below, function in general not influenced by the configuration
supported CPU mode	System Mode, User Mode and Firmware Mode	refer to the description of the CPU mode separation below below, function in general not influenced by the configuration

6.2 Security Assurance Requirements

Table 19 below lists all security assurance components that are valid for this Security Target. With two exceptions these security assurance components are required by EAL6 (see Section 2.2) or by the PP “Security IC Platform Protection Profile” [6].

The exceptions are the components ASE_TSS.2 and ALC_FLR.1. ASE_TSS.2 is chosen as an augmentation in this Security Target to give architectural information on the security functionality of the TOE. ALC_FLR.1 is chosen as an augmentation in this Security Target to cover policies and procedures of the developer applied to track and correct flaws and support surveillance 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 “EAL6 / PP” denotes, that an SAR is required by both EAL6 and the requirement of the PP [6], “EAL6” means that this requirement is due to EAL6 and beyond the requirement of the PP [6], and “PP” identifies this component as a requirement of the PP which is beyond EAL6. The augmentations ASE_TSS.2, ALC_FLR.1 chosen in this Security Target are denoted by "ST". The refinements of the PP [6], which must be adapted for EAL6, are described in section 6.2.1.

Table 19. Security Assurance Requirements

SAR	Title	Required by
ADV_ARC.1	Security architecture description	EAL6 / PP
ADV_FSP.5	Complete semi-formal functional specification with additional error information	EAL6
ADV_IMP.2	Complete mapping TDS vs. Implementation	EAL6
ADV_INT.3	Entire design well structured	EAL6
ADV_TDS.5	Semiformal description of all modules	EAL6
ADV_SPM.1	Formal model of Security Policies	EAL6
AGD_OPE.1	Operational user guidance	EAL6 / PP
AGD_PRE.1	Preparative procedures	EAL6 / PP
ALC_CMC.5	Production support, acceptance procedures and automation	EAL6
ALC_CMS.5	Development tools CM coverage	EAL6
ALC_DEL.1	Delivery procedures	EAL6 / PP
ALC_DVS.2	Sufficiency of security measures	PP
ALC_FLR.1	Basic flaw remediation	ST
ALC_LCD.1	Developer defined life-cycle model	EAL6 / PP
ALC_TAT.3	Standards used by 3rd party providers	EAL6
ASE_CCL.1	Conformance claims	EAL6 / PP
ASE_ECD.1	Extended components definition	EAL6 / PP
ASE_INT.1	ST introduction	EAL6 / PP
ASE_OBJ.2	Security objectives	EAL6 / PP
ASE_REQ.2	Derived security requirements	EAL6 / PP
ASE_SPD.1	Security problem definition	EAL6 / PP
ASE_TSS.2	TOE summary specification with architectural design summary	ST
ATE_COV.3	All interfaces completely tested	EAL6 / PP
ATE_DPT.3	Testing: modular design	EAL6
ATE_FUN.2	Functional testing, Analysis of testsequence	EAL6
ATE_IND.2	Independent testing - sample	EAL6 / PP
AVA_VAN.5	Advanced methodical vulnerability analysis	PP

6.2.1 Refinements of the Security Assurance Requirements

The Security Target claims conformance to the PP [6] and therefore it has to conform to the refinements of the TOE security assurance requirements (see Application Note 22 in [6]). Because the refinements in the PP [6] are defined for the security assurance components of EAL4, some refinements are necessary for assurance components of the higher level EAL6 claimed in the Security Target.

Table 20 lists the refinements of the PP [6] for the Security Target. Most of the refined security assurance components have the same level in both documents (PP [6] and Security Target). The following five subsections apply the refinements to ALC_CMS.5, ALC_CMC.5, ADV_FSP.5, ADV_IMP.2 and ATE_COV.3 which are different for the PP [6] and the Security Target.

Table 20. Security Assurance Requirements, overview of differences of refinements

Refined in PP [6]	Effect on Security Target
ALC_DEL	Same as in PP, refinement valid without change
ALC_DVS	Same as in PP, refinement valid without change
ALC_CMS	ALC_CMS.5, refinements valid without change
ALC_CMC	ALC_CMC.5, refinement valid without change
ADV_ARC	Same as in PP, refinement valid without change
ADV_FSP	ADV_FSP.5, refinements have to be adapted
ADV_IMP	ADV_IMP.2, refinement valid without change
ATE_COV	ATE_COV.3, refinement valid without change
AGD_OPE	Same as in PP, refinement valid without change
AGD_PRE	Same as in PP, refinement valid without change
AVA_VAN	Same as in PP, refinement valid without change [54]

The further Security Assurance Requirements especially the further augmentations added in this Security Target compared with the Protection Profile supplement and extent the Security Assurance Requirements and can be added without contradictions.

6.2.1.1 Refinements regarding CM scope (ALC_CMS)

This Security Target requires a higher evaluation level for the CC family ALC_CMS, namely ALC_CMS.5 instead of ALC_CMS.4. The refinement of the PP [6] regarding ALC_CMS.4 is a clarification of the configuration item “TOE implementation representation”. Since in ALC_CMS.5, the content and presentation of evidence element ALC_CMS.5.1C only adds an additional configuration item to the list of items to be tracked by the CM system, the refinement can be applied without changes.

The refinement of the configuration item “TOE implementation representation” of ALC_CMS.4 can be found in section the PP 6.2.1.3 of [6] and is not quoted here.

6.2.1.2 Refinements regarding CM capabilities (ALC_CMC)

This Security Target requires a higher evaluation level for the CC family ALC_CMC, namely ALC_CMC.5 instead of ALC_CMC.4. The refinement of the PP [6] regarding ALC_CMC.4 is a clarification of the “TOE” and the term "configuration items". Since in

54 According to Application Note 30 in the PP [6] the Security Target should indicate the version of the document [5] used for the vulnerability analysis. The current version is given in the bibliography.

ALC_CMC.5 requires a higher assurance regarding the defined TOE and the configuration items, the refinement can be applied without changes.

The refinements of the terms “TOE” and "configuration items" of ALC_CMC.4 can be found in section the PP 6.2.1.4 of [6] and is not quoted here.

6.2.1.3 Refinements regarding functional specification (ADV_FSP)

This Security Target requires a higher assurance level for the CC family ADV_FSP, namely ADV_FSP.5 instead of ADV_FSP.4. The refinement of the PP [6] regarding ADV_FSP.4 addresses the complete representation of the TSF, the purpose and method of use of all TSFI, and the accuracy and completeness of the SFR instantiations. The refinement is not a change in the wording of the action elements, but a more detailed definition of the items above is applied.

The higher level ADV_FSP.5 requires a Functional Specification in a “semi-formal style" (ADV_FSP.5.2C).

The component ADV_FSP.5 extends the scope of the error messages to be described from those resulting from an invocation of a TSFI (ADV_FSP.5.6C) to also those not resulting from an invocation of a TSFI (ADV_FSP.5.7C). For the latter a rationale shall be provided (ADV_FSP.5.8C).

Since the higher level ADV_FSP.5 only affects the style of description and the scope of and rationale for error messages, the refinements can be applied without changes and are valid for ADV_FSP.5.

The refinement of the original component ADV_FSP.4 can be found in Section 6.2.1.6 of the Protection Profile [6] and is not quoted here.

6.2.1.4 Refinements regarding Implementation Representation (ADV_IMP.2)

The refinement of the PP [6], section 6.2.1.7, regarding Implementation Representation states that the provided implementation representation is complete and sufficient to ensure that analysis activities are not curtailed due to lack of information. This Security Target requires assurance level EAL6 augmented which requires access to complete source code. Therefore the refinement of the PP [6] is implicitly fulfilled.

6.2.1.5 Refinements regarding Test Coverage (ATE_COV.3)

Both refinements defined in the PP [6], section 6.2.1.8, regarding Test Coverage are applied without change. The refinements defining test coverage under different operating conditions and proof of existence and effectiveness of countermeasures against physical attacks. As this Security Target requires assurance level EAL6 augmented the latter refinement can be applied without change and can be fulfilled by access to complete source code and layout data. The refinement of test coverage under different operating conditions is not a change in the wording of the action elements, but a more detailed definition of the items above to be applied and therefore can be applied without changes.

6.2.2 Definition of ADV_SPM

The developer shall provide a formal security policy model for the *Access Control Policy*.[55]

The *Access Control Policy* comprises the following Security Functional Requirements: FDP_ACC.1[MEM], FDP_ACC.1[SFR], FDP_ACF.1[MEM], FDP_ACF.1[SFR] with the associated dependencies. Further the secure state as required by FDP_FLS.1 is

[55] [assignment: list of policies that are formally modeled].

included in the security policy model. In addition parts of the life cycle control as required by FMT_LIM.2 are part of the model.

6.3 Security Requirements Rationale

6.3.1 Rationale for the security functional requirements

Section 6.3.1 in the PP [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 21. Security Requirements versus Security Objectives

Objective	TOE Security Functional Requirements
O.Leak-Inherent	FDP_ITT.1[HW] “Basic internal transfer protection”
	FPT_ITT.1[HW] “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[HW], FPT_ITT.1[HW], 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[HW], FPT_ITT.1[HW], FDP_IFC.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1
O.Identification	FAU_SAS.1[HW] “Audit storage”
O.RND	FCS_RNG.1[HW] “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[HW], FPT_ITT.1[HW], FDP_IFC.1, FPT_PHP.3, FRU_FLT.2, FPT_FLS.1

The Security Target extends SFR defined in the PP [6] and additionally defines SFRs as listed in Table 22. The following table gives an overview, how the requirements are combined to meet the security objectives.

Table 22. Mapping of security objectives and requirements

Objective	TOE Security Functional Requirement
O.INTEGRITY_CHK	FDP_ITT.1[HW] “Basic internal transfer protection”
	FPT_ITT.1[HW] “Basic internal TSF data transfer protection”
	The SFR of the PP are extended regarding manipulation. The same information flow control policy as defined in the PP applies.
O.HW_DES3	FCS_COP.1[HW_DES]
O.HW_AES	FCS_COP.1[HW_AES]

Objective	TOE Security Functional Requirement
O.FM_FW	FDP_ACC.1[MEM, FDP_ACF.1[MEM], FMT_MSA.3[MEM]
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]
O.SFR_ACCESS	FDP_ACC.1[SFR], FDP_ACF.1[SFR], FMT_MSA.3[SFR], FMT_MSA.1[SFR], FMT_SMF.1[HW]
O.CUST_RECONFIG	FMT_SMF.1[HW]
O.EEPROM_INTEGRITY	FDP_SDI.2[HW]

The justification related to the security objective "Integrity control of transferred data" (O.INTEGRITY_CHK) is as follows:

O.INTEGRITY_CHK requires the TOE to check the integrity of user data and TSF data if transferred between different parts of the TOE. Exactly this is the extended requirement of FDP_ITT.1[HW] and FPT_ITT.1[HW] compared to the PP [6]. Therefore FDP_ITT.1[HW] and FPT_ITT.1[HW] are suitable to meet O.INTEGRITY_CHK.

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 security objective “Firmware Mode Firewall” (O.FM_FW) is as follows:

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

The security functional requirement “Static attribute initialisation (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 generated by the reset procedure and the Boot-ROM Software for the related Special Function Register. Restrictive with respect to memory partition means that the partition cannot be changed at all and for the memory segmentation means that the initial setting is very restrictive since it effectively disables any memory segment. They are needed by the TOE to provide a default configuration after reset. 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 that

can be used to change the memory partition. Also no related management function is specified by FMT_SMF.1[HW]. Therefore the memory partition is fixed and cannot be changed any subject, which is the requirement of O.FM_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) “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 initialisation (FMT_MSA.3[MEM])” requires that the TOE provide default values for the security attributes used by the Memory Management Units. Since the TOE is a hardware platform these default values are generated by the reset procedure for the related Special Function Register. They are needed by the TOE to provide a default configuration after reset. 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 realised 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 required 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) “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 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 System Mode as specified by the Security Function Policy (SFP) “Access Control Policy”. In 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 require certain security attributes to implement the access control to Special Function Register as required 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 initialisation (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 realised in a way that – besides the definition of access rights to Special Function Registers related to hardware components in User Mode and Firmware 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[HW])” 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 are needed to separate the different types of objects because they have different security attributes.

The justification related to the security objective “Integrity support of data stored in EEPROM” (O.EEPROM_INTEGRITY) is as follows:

The security functional requirement “Stored data integrity monitoring and action (FDP_SDI.2[HW])” is used for the specification of the control function to adjust the conditions of an EEPROM block such that the integrity of the data read from EEPROM is ensured even if the characteristics of the EEPROM changed e.g. due to ageing. Therefore, FDP_SDI.2[HW] is suitable to meet the security objective.

The justification related to the security objective “Customer Option Reconfiguration” (O.CUST_RECONFIG) is as follows:

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

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 of the PP [6] are fulfilled within the PP [6] and at least one dependency is considered to be satisfied.

The following discussion demonstrates how the dependencies defined by Part 2 of the Common Criteria [2] for the 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:

Table 23. 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	See discussion below
	FCS_CKM.4	
FCS_COP.1[HW_AES]	FDP_ITC.1 or FDP_ITC.2, or FCS_CKM.1	See discussion below
	FCS_CKM.4	
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	Yes, by FDP_ACC.1[MEM]
	FMT_MSA.3	Yes, by FMT_MSA.3[MEM]
FDP_ACF.1[SFR]	FDP_ACC.1	Yes, by FDP_ACC.1[SFR]
	FMT_MSA.3	Yes, by FMT_MSA.3[SFR]
FMT_MSA.3[MEM]	FMT_MSA.1	Yes, by FMT_MSA.1[MEM]
	FMT_SMR.1	See discussion below
FMT_MSA.3[SFR]	FMT_MSA.1	Yes, by FMT_MSA.1[SFR]
	FMT_SMR.1	See discussion below
FMT_MSA.1[MEM]	FDP_ACC.1 or FDP_IFC.1	Yes, by FDP_ACC.1[MEM]
	FMT_SMR.1	See discussion below
	FMT_SMF.1	Yes, by FMT_SMF.1[HW]
FMT_MSA.1[SFR]	FDP_ACC.1 or FDP_IFC.1	Yes, by FDP_ACC.1[SFR]
	FMT_SMR.1	See discussion below
	FMT_SMF.1	Yes, by FMT_SMF.1[HW]

The developer of the Security IC Embedded Software must ensure that the additional security functional requirements FCS_COP.1[HW_DES] and FCS_COP.1[HW_AES] are used as specified and that the User Data processed by the related security functionality is protected as defined for the application context.

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 as specified for the memory access control function. All requirements concerning these management functions shall be fulfilled by the environment (Security IC Embedded Software).

The functional requirements [FDP_ITC.1, or FDP_ITC.2 or FCS_CKM.1] and FCS_CKM.4 are not included in this Security Target since the TOE only provides a pure engine for encryption and decryption without additional features for the handling of cryptographic keys. Therefore the Security IC Embedded Software must fulfill these requirements related to the needs of the realised application.

The dependency FMT_SMR.1 introduced by the two 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.

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, but chooses a higher assurance level. The level EAL6 is chosen in order to meet assurance expectations of digital signature applications and electronic payment systems. Additionally, the requirement of the PP [6] to choose at least EAL4 is fulfilled.

The rationale for the augmentations is the same as in the PP. The assurance level EAL6 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 6. Therefore, these components add additional assurance to EAL 6, but the mutual support of the requirements is still guaranteed.

As stated in the Section 6.3.3 of the PP [6], it has to be assumed that attackers with high attack potential try to attack smartcards 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 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, the integrity support of data stored in EEPROM and the memory access/separation control function as well as the access control to Special Function Register implemented according to the security functional requirement FCS_COP.1[HW_DES], FCS_COP.1[HW_AES], FDP_SDI.2[HW] and FDP_ACC.1[MEM], FDP_ACC.1[SFR] with reference to the Access Control Policies defined in FDP_ACF.1[MEM] and FDP_ACF.1[SFR]. Therefore, these security functional requirements support the secure implementation and operation of FCS_COP.1[HW_DES], FCS_COP.1[HW_AES] and of FDP_ACC.1 with FDP_ACF.1 as well as the dependent security functional requirements.

The extension of the security functional requirements FDP_ITT.1[HW] and FTP_ITT.1[HW] compared to the underlying PP [6] adds the detection of faults during the transfer of user data or TSF data between internal components of the TOE. The protection against leakage is not weakened by this extension.

A hardware platform including the Security IC Dedicated Software requires Security IC Embedded Software to build a secure product. Thereby the Security IC Embedded Software must support the security functionality of the hardware as well as the Security IC Dedicated Support Software and implement a sufficient management of the security services implemented by the hardware platform including the Security IC Dedicated Software. The realisation of the Security Functional Requirements within the TOE provides a good balance between flexible configuration and restrictions to ensure a secure behaviour 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 Security Functionality provided by the TOE is split into Security Services (SS) and Security Features (SF). Both are active and 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 security functionality is active after phase 3 or phase 4 depending on the delivery form.

The TOE also comprises security mechanisms, which are not listed as security functionality in the following. Such mechanisms do not implement a complete Security Services or Security Features. They can be used to implement further Security Services and/or Security Features based on Security IC Embedded Software using these security mechanisms, e.g. the Fame2 coprocessor for asymmetric cryptographic algorithms.

7.1.1 Security Services

SS.RNG: Random Number Generator

The Random Number Generator continuously produces random numbers with a length of one byte. The TOE implements SS.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 hardware detects or bad quality of the produced random numbers.

According to AIS31 [7] the Random Number Generator claims the functionality class PTG2. 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, generation of seeds for DRNGs and fulfils the online test requirements defined in [7].

SS.HW_DES: Triple-DES coprocessor

The TOE provides the Triple Data Encryption Algorithm (TDEA) according to the Data Encryption Standard (DES) [18]. 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 [18]. 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.

SS.HW_DES also supports hardware XOR-operation of two data blocks to support chaining modes of the TDES if this is configured 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 [19]. 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. SS.HW_AES also supports hardware XOR-operation of two data blocks to support chaining modes of the AES if this is configured by the Security IC Embedded Software.

SS.CRC: Cyclic Redundancy Check

The TOE provides a 16- and 32-bit Cyclic Redundancy Calculation (CRC) checksum calculation mechanism. SS.CRC is a modular checksum calculation function, which provides checksum calculation supporting the CRC polynomials 0x1021 (CCITT,X25) for 16-bit checksum calculation and 0x04C11DB7 for 32-bit checksum calculation, which must be selected by the Security IC Embedded Software. In combination with the memory verify functionality of the CPU SS.CRC provides a mechanism for checksum calculation over memory segments applying memory access control mechanisms according to SF.MEM_ACC without reading these memory segments by the Security IC Embedded Software. If SS.CRC is used in combination with the copy machine the CRC is calculated during the transfer of the data between different parts of the TOE. SS.CRC is used in Boot Mode by the IC dedicated Software for calculation of a CRC checksum during the transfer of the Security Rows content into Special Function Registers.

SS.RECONFIG: Post Delivery Configuration

SS.RECONFIG realizes the Post Delivery Configuration. These can be used by the customer to set the accessible size of the EEPROM and the accessible size of the CXRAM and to enable or disable the Fame2 coprocessor, the AES coprocessor and the contactless interface. The configuration values of the Post Delivery Configuration are stored in a special area in the Security Row (see SF.COMP).

Note that if the Fame2 coprocessor and the AES coprocessor are disabled, both will no longer be available to the Security IC Embedded Software and attempting to use it will raise an exception. This means the availability of SS.HW_AES is configurable.

The customer can change the values of the Post Delivery Configuration through invoking the Post Delivery Configuration functionality in the Boot-ROM Software (see SF.MEM_ACC). This functionality is invoked by using the chip health mode via the ISO/IEC 7816 interface and applying the required Post Delivery Configuration commands.

The customer can change these values as many times as he wishes. However, once he calls the Boot-ROM Software using the chip health mode via the ISO/IEC 7816 interface with a certain parameter set to a specific value, the options are locked permanently, and can no longer be changed. The options must be locked before the TOE is delivered to the customer before phase 7 of the life cycle.

7.1.2 Security Features

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 Fame2 coprocessor, the memories, registers, I/O interfaces and the other system peripherals) during execution of the Security IC Dedicated Support Software and Security IC Embedded Software. This includes all security mechanisms of the TOE, which directly contribute to a Security Service or a Security Feature.

The TOE ensures its correct operation and prevents any malfunction using the following mechanisms: filtering of power supply, clock frequency and reset input as well as monitoring of voltage supply, clock frequency input and the temperature of the chip by means of sensors. There are multiple voltage and frequency 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. In addition to the light sensors the EEPROM provides two functions to detect light attacks. The Security IC Embedded Software can enable/disable the EEPROM double read function.

Specific functional units of the TOE are equipped with further fault injection detection. This comprises the Secure Fetch for the processing of code and data by the CPU or special circuitry within a functional unit. The functional units are the Program Counter, the stack pointer, the CPU control register, the MMU address cache and control registers, the SBC interface for the DES and the AES coprocessors, the Fame2 coprocessor, Copy Machine control registers and hardware configuration as well as test control registers. Furthermore the TOE contains a watchdog timer which can be enabled and configured by the Security IC Embedded Software to protect the program execution.

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. In case minor configuration option “Inverse EEPROM Error Correction” is enabled (see Section 1.4.2.2) the probability to detect fault injection errors at the EEPROM memory and interface increases and the error correction logic raises an exception when detecting an error. The RAM memory is equipped with additional parity bits which are checked when the corresponding data stored in RAM is read. Additionally a RAM parity watchdog mechanism is supported which checks the parity bits of the complete RAM in random order when the Security IC Embedded Software is not accessing the RAM. In case of detected RAM parity errors both mechanisms trigger a security reset. In case the TOE processes a reset all components of the TOE are initialised 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. The TOE defends the sensors from being disabled by the Security IC Embedded Software.

The TOE controls the specified range of the stack pointer. The stack pointer and the related control logic are implemented threefold for the User Mode, System Mode and Super System Mode (comprising Boot Mode, Test Mode and Firmware 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 Security 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 Security Rows. 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 the TOE and specific

encryption mechanisms for the memories and internal buses. SF.PHY supports the efficiency of other portions of the security functionality.

SF.PHY also supports the integrity of the EEPROM, RAM and the ROM. The EEPROM is able to correct a 1-bit error within each byte. The EEPROM corrects these errors automatically without user interaction. The RAM and the ROM provide a parity check. In both cases a reset is forced based on a parity error.

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, signals on other pads, which are not intentionally used for communication by the terminal or the Security IC Embedded Software as well as emanation of the hardware platform. 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 or emanation and subsequent complex signal analysis. This protection of the TOE is enforced by several security mechanisms in the design, which support these portions of security functionality.

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

The AES coprocessor includes specific security mechanisms to prevent SPA/DPA analysis of shape and amplitude of the power consumption and emanation. The implementation of the AES coprocessor further ensures that the calculation time is independent from the chosen key any plain/cipher text for a given key length.

The Fame2 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 Fame2 coprocessor does not realise an algorithm on its own and algorithm-specific leakage countermeasures have to be added by the Security IC Embedded Software when using the Fame2 coprocessor.

Additional security mechanisms can be configured by the Security IC Embedded Software. This comprises the configuration of the clock that can be used to prevent the synchronisation between internal operations and external clock or 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) 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) Firmware Mode. This includes protection of electronic fuses stored in a protected memory area, the so-called Security Rows, and the possibility to store initialisation or pre-personalisation data in the so-called FabKey Area.

Control of the CPU mode for Boot Mode, Test Mode and Firmware Mode prevents 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 integrity control of electronic fuses ensures secure storage and setup of configuration and calibration data, during the start-up in Boot Mode. The protection of electronic fuses especially ensures that configuration options with regard to the security functionality can not be changed, abused or influenced in any way. The transfer of the content of the electronic fuses into the related hardware configuration registers is protected by a CRC checksum generated using SS.CRC. SF.COMP ensures that activation or deactivation of security mechanisms can not be influenced by the Security IC Embedded Software so that the TSF provides self-protection against interference and tampering by untrusted Security IC Embedded Software.

SF.COMP protects CPU mode switches regarding Boot Mode, Test Mode and Firmware Mode in the following way: Switching from Boot Mode to Test Mode or Firmware 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 can not be invoked afterwards. Therefore, once the TOE has left the test phase and each time the TOE completed start-up or reset, Firmware Mode is the only Super System Mode available.

The TSF controls access to the Security Rows, the top-most 512 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, which 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’.

SF.COMP also provides a memory area in the Security Row where the current values for the Post Delivery Configuration (see SS.RECONFIG). This area cannot be accessed by the Security IC Embedded Software, but it can be accessed by the Resource Configuration Firmware.

If minor configuration option “Card Disable Function" is used (refer to section 1.4.2.2) SF.COMP inhibits any start-up of the Security IC Embedded Software once the Security IC Embedded Software disables the card.

If minor configuration option “EEPROM application content erase” is used (see Section 1.4.2.2) SF.COMP erases the application data stored in the EEPROM once the Security IC Embedded Software has activated this security feature.

SF.COMP also provides the FabKey Area where initialisation 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.

The complete EEPROM is initialized during wafer testing and pre-personalisation. The values for the Security Row depend on the configuration options and choice of the Security IC Embedded Software developer. The values for the application EEPROM 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 personalised) 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-personalisation 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 Security 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. Access control is conducted in two ways:

- Memory partitioning: Each memory type ROM, RAM and EEPROM is partitioned into two parts. In Boot Mode and Firmware Mode the CPU has access to the Firmware-EEPROM, Firmware-RAM and Test-ROM. In System Mode and User Mode the CPU has access to the Application-EEPROM, Application-RAM and Application-ROM. 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 memory 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 executed in User Mode.

Memory partitioning is fixed and can not be changed. It is determined during production of the TOE and is solely dependent on the major configuration and the minor configuration option OS Emulation (see Section 1.4.2.2) and Post Delivery Configuration (see Section 1.4.2.3).

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 can not 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. Any access outside the boundaries of the physical implemented memory in all CPU modes except the Test Mode and access to forbidden memory addresses in User Mode are notified by raising an exception.

The Memory Management Unit also handles access rights to Special Function Registers related to hardware components for code running in Firmware Mode. The configuration of the access rights for User Mode and Firmware Mode are used by SF.SFR_ACC to grant or block access to the related Special Function Registers. The access rights can be defined for up to 16 groups of Special Function Registers, which are related to 16 hardware peripherals or memories described in [9], section 12.4. Thus, User Mode and Firmware Mode can be restricted in their access to the Special Function Registers related to 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 specific Special Function Register.

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. Key registers cannot be read since they are write-only to support the confidentiality of keys. 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 controls accesses to Special Function Registers. If the access is not allowed or the Special Function Register addressed by the code is not implemented an exception is triggered. The Security IC Embedded Software can react on this exception.

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 Firmware Mode. Hence, such Special Function Registers are inherently separated and enforce the separation between the CPU modes.

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 can not be configured by the code running on the hardware platform.

This implies that code running in User Mode or Firmware Mode is not able to use SS.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 specific Special Function Register.

The CPU mode changes by the following operations.

- Call of a system call vector (SVEC) address or a firmware vector (FVEC) address. A call of a SVEC enables System Mode, a call of a FVEC sets enables Firmware Mode. Calls of SVEC addresses are allowed in User Mode only, otherwise an exception is raised.
- Execution of an exception or interrupt. Any event that leads to the execution of an exception leads to a special status of the related CPU mode. Any further exception in this special status forces a reset. Interrupts can be executed in User Mode or in System Mode as well as in Firmware Mode. The Security IC Embedded Software running in System Mode can configure this option at run time.
- Return from an exception/interrupt or vector call with a RETI instruction. This restores the CPU mode 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 FVEC or SVEC call, but no return address is pushed onto the stack.
- Direct modification of the specific Special Function Register. 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 related to 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 in regard to 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 Security IC Dedicated Software. As described above, SF.MEM_ACC provides the access control information to Special Function Registers related to hardware components in Firmware Mode and User Mode.

SF.FFW: Firmware Firewall

SF.FFW (Protected Firmware Mode) implements a mechanism to protect the application data of the different firmware applications (NXP firmware functionality) running in Firmware Mode by means of a software firewall separating the application data between each other.

The software firewall mechanism is based on the security features SF.MEM_ACC and SF.SFR_ACC.

SF.FIRMWARE: Firmware Support

SF.FIRMWARE (Firmware Operating System) implements specific basic support functionality for the Security IC Embedded Software. The basic support functionality is implemented in a way that the protection and separation of the different type of User

Data is enforced. The security feature SF.FIRMWARE is based on the security features SF.MEM_ACC, SF.SFR_ACC and SF.FFW.

The provided functionality comprises the integrity protection of the data stored in the EEPROM and resource reconfiguration support. Additionally the baud rate configuration for the contactless operation is supported.

7.2 TOE Summary Specification Rationale

7.2.1 Mapping of Security Functional Requirements and 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.

Table 24. Mapping of Security Functional Requirements and the portions of the TOE Security Functionality

	SS.RNG	SS.HW_DES	SS.HW_AES	SS.CRC	SF.OPC	SF.PHY	SF.LOG	SF.COMP	SF.MEM_ACC	SF.SFR_ACC	SF.FFW	SF.FIRMWARE	SS.RECONFIG
FAU_SAS.1[HW]								X					
FCS_RNG.1[HW]	X												
FDP_IFC.1	O	O	O	O			X						
FDP_ITT.1[HW]	O	O	O	O			X						
FPT_ITT.1[HW]	O	O	O	O			X						
FMT_LIM.1								X					
FMT_LIM.2								X	X	X			
FPT_FLS.1		O	O		X			O	O	O			
FRU_FLT.2					X								
FPT_PHP.3						X							
FCS_COP.1[HW_DES]		X											
FCS_COP.1[HW_AES]			X										
FDP_ACC.1[MEM]									X		X		
FDP_ACC.1[SFR]										X	X		
FDP_ACF.1[MEM]									X		X		
FDP_ACF.1[SFR]										X	X		
FMT_MSA.1[MEM]									X		X		
FMT_MSA.1[SFR]										X	X		

	SS.RNG	SS.HW_DES	SS.HW_AES	SS.CRC	SF.OPC	SF.PHY	SF.LOG	SF.COMP	SF.MEM_ACC	SF.SFR_ACC	SF.FFW	SF.FIRMWARE	SS.RECONFIG
FMT_MSA.3[MEM]									X		X		
FMT_MSA.3[SFR]										X	X		
FMT_SMF.1[HW]									X	X	X		X
FDP_SDI.2[HW]												X	

"X" in the table above means that the specific portion of the TOE security functionality realises the functionality required by the respective Security Functional Requirement. “O” in the table above means that the specific portion of the TOE security functionality supports the functionality required by the respective Security Functional Requirement.

As already stated in the definition of the portions of the TOE security functionality there are additional security mechanisms, which can contribute to security functionality when they are appropriately controlled by the Security IC Embedded Software. E.g. the Fame2 coprocessor can be used to implement leakage-resistant asymmetric cryptographic algorithms.

As part of the Security IC Dedicated Software the NXP firmware functionality is separated from the Security IC Embedded Software by memory partitioning according to SF.MEM_ACC, SF.FFW and by CPU mode control according to SF.SFR_ACC and SF.COMP. The data exchange areas are also controlled by SF.MEM_ACC and SF.FFW and must be configured by the Security IC Embedded Software. This ensures that the OS Emulation functionality does not violate the TSF.

7.2.2 Rationale for the portions of the TOE security functionality

(deleted here, only available in the full version of the Security Target)

7.2.3 Security architectural information

Since this Security Target claims the assurance requirement ASE_TSS.2 security architectural information on a very high level is supposed to be included in the TSS to inform potential customers on how the TOE protects itself against interference, logical tampering and bypass. In the security architecture context, this covers the aspects self-protection and non-bypassability.

(details deleted here, available only in the full version of the Security Target)

8. Annexes

8.1 Further Information contained in the PP

Chapter 7 of the PP “Security IC 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] contains 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 finalising the IC Embedded Software and the applications on platform transforming the TOE into the unpersonalised 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 Personaliser (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, Firmware Mode, System Mode and User Mode.
DESFire	DESFire EV1 emulation, names the DESFire Operating System as part of the Security IC Dedicated Software.
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, unauthorised 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.
Firmware Mode	CPU mode of the TOE dedicated to execution of the Emulation Framework, MIFARE DESFire and MIFARE Plus Operating System, which is part of the Security IC Dedicated Support Software. This mode is not accessible for the Security IC Embedded Software.
End-consumer	User of the Composite Product in Phase 7
Initialisation Data	Initialisation 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).
Integrated Circuit (IC)	Electronic component(s) designed to perform processing and/or memory functions.
kByte(s) / (KB)	kByte (KB) used with k=1024 (K=1024)
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 Plus	MIFARE Plus emulation, names the MIFARE Plus Operating System as part of the Security IC Dedicated 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-personalisation Data	Any data supplied by the Card Manufacturer that is injected into the non-volatile memory by the Integrated Circuits manufacturer (Phase 3). These data are for instance used for traceability and/or to secure shipment between phases.
S^2C	Smartcard 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 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 (Security IC Dedicated Test Software) but may provide additional services to facilitate usage of the hardware and/or to provide additional services (Security IC Dedicated Support Software).
Security IC Dedicated Support Software	That part of the Security IC Dedicated Software (refer to above) which provides functions after TOE Delivery. The usage of parts of the Security IC Dedicated Software might be restricted to certain phases.

Security IC Dedicated Test Software	That part of the Security IC Dedicated Software (refer to above) which is used to test the TOE before TOE Delivery but which does not provide any functionality thereafter.
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 Security 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 256 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 Fame2 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 Firmware 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 Security 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 Security 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 242H10) 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 243H1.2.2) and its development and production environment are fulfilled (refer to [6], Figure 2 on page 24H10). 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
CPU_CSR	CPU Status Register
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
PDC	Post Delivery Configuration
PKC	Public Key Cryptography
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	as abbreviation of the CC term: Security Functional Requirement, as abbreviation of the technical term of the SmartMX2-family: Special Function Register[56]
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

[56] 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 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 for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 3, July 2009, CCMB-2009-07-002

[3] Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, 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] Supporting Document Mandatory Technical Document Application of Attack Potential to Smartcards, Version 2.7, Revision 1, March 2009, CCDB-2009-03-001

[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] Anwendungshinweise und Interpretationen zum Schema, AIS31: Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren, Version 2.1, 2.12.2011, Bundesamt für Sicherheit in der Informationstechnik

[8] A proposal for: Functionality classes for random number generators, Version 2.0, 18 September 2011

9.1.2 Developer Documents

[9] Product Data Sheet SmartMX2 family P60D012/016/024 Secure high-performance smart card controller, NXP Semiconductors

[10] Instruction Set for the SmartMX2 family, Secure smart card controller, NXP Semiconductors, Business Unit Identification

[11] Guidance and Operation Manual NXP Secure Smart Card Controller P60D024/016/012 VB, NXP Semiconductors, Business Unit Identification

[12] SmartMX2 P60D012/016/024 VB Wafer and delivery specification, NXP Semiconductors, Business Unit Identification

[13] Order Entry Form P60D024, NXP Semiconductors, Business Unit Identification, online document

[14] Order Entry Form P60D016, NXP Semiconductors, Business Unit Identification, online document

[15] Order Entry Form P60D012, NXP Semiconductors, Business Unit Identification, online document

[16] Product data sheet addendum: SmartMX2 family, Post Delivery Configuration (PDC), NXP Semiconductors

[17] Product data sheet addendum: SmartMX2 family Chip Health Mode (CHM), NXP Semiconductors

[18] Impact Analysis Report NXP Secure Smart Card Controller P60D024/016/012PVB, NXP Semiconductors, Business Unit 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] Mifare Interface Platform, V2.11, Philips Semiconductors, BL Identification

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

Limited warranty and liability — 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.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the Terms and conditions of commercial sale of NXP Semiconductors.

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 at 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.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on a weakness or default in the customer application/use or the application/use of customer's third party customer(s) (hereinafter both referred to as "Application"). It is customer's sole responsibility to check whether the NXP Semiconductors product is suitable and fit for the Application planned. Customer has to do all necessary testing for the Application in order to avoid a default of the Application and the product. NXP Semiconductors does not accept any liability in this respect.

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 Trademarks

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

DESFire — is a trademark of NXP B.V.

FabKey — is a trademark of NXP B.V.

MIFARE — is a trademark of NXP B.V.

MIFARE Plus — is a trademark of NXP B.V.

SmartMX — is a trademark of NXP B.V.

.

11. List of figures

Fig 1. Block Diagram of P60D024/016/012PVB.....5

12. List of tables

Table 1. Components of the TOE 6
Table 2. Evaluated major configuration options 8
Table 3. Evaluated major configuration options 8
Table 4. Evaluated minor configuration options 8
Table 5. Post Delivery Configuration for P60D024/016/012PVB 10
Table 6. Variable definitions for commercial type names 11
Table 7. Supported Package Types 12
Table 8. CPU modes of the TOE 12
Table 9. Threats defined by the PP [6] 21
Table 10. Additional threats averted by the TOE 22
Table 11. Assumptions defined in the PP [6] 23
Table 12. Security objectives defined in the PP [6] 25
Table 13. Security objectives for the Security IC Embedded Software development environment, taken from the PP [6] 27
Table 14. Security objectives for the operational environment, taken from the PP [6] 27
Table 15. Security Objectives versus Assumptions, Threats or Policies 28
Table 16. Additional Security Objectives versus Assumptions, Threats or Policies 28
Table 17. SFRs taken from the PP [6] 32
Table 18. Differences between TOE configurations with regard to the Access Control Policy 48
Table 19. Security Assurance Requirements 49
Table 20. Security Assurance Requirements, overview of differences of refinements 50
Table 21. Security Requirements versus Security Objectives 52
Table 22. Mapping of security objectives and requirements 52
Table 23. Dependencies of security functional requirements 56
Table 24. Mapping of Security Functional Requirements and the portions of the TOE Security Functionality 66

13. Contents

1. ST Introduction 3
1.1 ST Reference 3
1.2 TOE Reference 3
1.3 TOE Overview 3
1.3.1 Usage and major security functionality of the TOE 3
1.3.2 TOE type 5
1.3.3 Required non-TOE hardware/software/firmware 5
1.4 TOE Description 5
1.4.1 Physical Scope of TOE 5
1.4.1.1 TOE components 6
1.4.2 Evaluated configurations 7
1.4.2.1 Major configuration options 7
1.4.2.2 Minor configuration options 8
1.4.2.3 Post Delivery Configuration 10
1.4.2.4 Evaluated package types 11
1.4.3 Logical Scope of TOE 12
1.4.3.1 Hardware Description 12
1.4.3.2 Software Description 15
1.4.3.3 Documentation 15
1.4.4 Security during Development and Production .. 16
1.4.5 TOE Intended Usage 16
1.4.6 Interface of the TOE 17
2. Conformance Claims 19
2.1 CC Conformance Claim 19
2.2 Package claim 19
2.3 PP claim 19
2.4 Conformance Claim Rationale 20
3. Security Problem Definition 21
3.1 Description of Assets 21
3.2 Threats 21
3.3 Organisational Security Policies 22
3.4 Assumptions 23
4. Security Objectives 25
4.1 Security Objectives for the TOE 25
4.2 Security Objectives for the Security IC Embedded Software development Environment 26
4.3 Security Objectives for the Operational Environment 27
4.4 Security Objectives Rationale 28
5. Extended Components Definition 31
6. Security Requirements 32
6.1 Security Functional Requirements 32
6.1.1 SFRs of the Protection Profile 32
6.1.2 Additional SFRs regarding cryptographic functionality 36
6.1.3 Additional SFRs regarding access control 38
6.2 Security Assurance Requirements 48
6.2.1 Refinements of the Security Assurance Requirements 50
6.2.1.1 Refinements regarding CM scope (ALC_CMS) 50
6.2.1.2 Refinements regarding CM capabilities (ALC_CMC) 50
6.2.1.3 Refinements regarding functional specification (ADV_FSP) 51
6.2.1.4 Refinements regarding Implementation Representation (ADV_IMP.2) 51
6.2.1.5 Refinements regarding Test Coverage (ATE_COV.3) 51
6.2.2 Definition of ADV_SPM 51
6.3 Security Requirements Rationale 52
6.3.1 Rationale for the security functional requirements 52
6.3.2 Dependencies of security functional requirements 55
6.3.3 Rationale for the Assurance Requirements 57
6.3.4 Security Requirements are internally Consistent 57
7. TOE Summary Specification 58
7.1 Portions of the TOE Security Functionality 58
7.1.1 Security Services 58
7.1.2 Security Features 59
7.2 TOE Summary Specification Rationale 66
7.2.1 Mapping of Security Functional Requirements and TOE Security Functionality 66
7.2.2 Rationale for the portions of the TOE security functionality 67
7.2.3 Security architectural information 67
8. Annexes 68
8.1 Further Information contained in the PP 68
8.2 Glossary and Vocabulary 68
8.3 List of Abbreviations 73
9. Bibliography 74
9.1.1 Evaluation Documents 74
9.1.2 Developer Documents 74
9.1.3 Other Documents 75
10. Legal information 76
10.1 Definitions 76
10.2 Disclaimers 76
10.3 Licenses 76

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

Date of release: 04 November 2013
Document identifier:

10.4 Trademarks 76
11. List of figures 77
12. List of tables 78
13. Contents 79

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

Date of release: 04 November 2013
Document identifier: