NXP Secure Smart Card Controller N7021 VA

Security Target Lite

Rev. 2.3 – 2019-06-04 **Evaluation documentation**

Final **Public**

BSI-DSZ-CC-0977

Document Information

Info	Content
Keywords	CC, Security Target Lite, N7021 VA
Abstract	Security Target Lite of the NXP Secure Smart Card Controller N7021 VA, which is developed and provided by NXP Semiconductors, Business Unit Security & Connectivity according to the Common Criteria for Information Technology Security Evaluation Version 3.1 at EAL6 augmented

Rev	Date	Description
1.0	03-April-2017	First version
1.1	31-May-2017	Minor update after certifier feedback.
2.0	06-September-2018	Updated document version numbers in Tab. 1.1. Updated CC conformance to v3.1 rev5.
2.1	15-November-2018	Updated SP 800-67 reference. Updated delivery information in section 1.4.1.1.
2.2	09-May-2019	Removed single-DES and 2-key TDES references.
2.3	06-June-2019	Updated Guidance and Operation Manual reference.

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 N7021 VA Security Target, 2.3, NXP Semiconductors, 2019-06-04.

1.2 TOE Reference

The TOE is named "NXP Secure Smart Card Controller N7021 VA including IC Dedicated Software". In this document the TOE is abbreviated to NXP Secure Smart Card Controller N7021 VA or to N7021 VA. All components of the TOE and their respective version numbers are listed in section 1.4.1.1.

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 N7021 VA with IC Dedicated Software and documentation describing instruction set and usage of the TOE. The TOE does not include a customer-specific Security IC Embedded Software.

The IC hardware is a microcontroller incorporating a central processing unit (CPU), memories accessible via a Memory Management Unit (MMU), cryptographic coprocessors, other security components and several electrical communication interfaces. The central processing unit supports a 32-/16-bit instruction set optimized for smart card applications. The first and in some cases the second byte of an instruction are used for operation encoding. On-chip memories are ROM, RAM and Flash. The Flash can be used as data or program memory. It consists of highly reliable memory cells, which are designed to provide data integrity. Flash is optimized for applications that require reliable non-volatile data storage for data and program code. Dedicated security functionality protects the contents of all memories. Notice, that the Flash is also referred to as Non-Volatile Memory (NVM) in this Security Target.

The IC Dedicated Software comprises IC Dedicated Test Software for test purposes and IC Dedicated Support Software. The IC Dedicated Support Software consists of the Boot Software, which controls the boot process of the hardware platform. Furthermore, it provides a Firmware Interface and optionally Shared OS Libraries, simplifying access to the hardware for the Security IC Embedded Software. A System Mode OS is available (optional), offering ready-to-use resource and access management for customer applications that do not want to be exposed to the more low-level features of the TOE. The Flashloader OS (optional) supports download of code and data to Flash by the Composite Product Manufacturer before Operational Usage (e.g. during development). The Symmetric Crypto Library (optional) provides simplified access to frequently used symmetric cryptography algorithms.

The documentation includes a Product Data Sheet with several addenda, an Instruction Set Manual, a Guidance and Operation Manual, Symmetric Crypto Library User Manuals and a Wafer and Delivery Specification. This documentation comprises a description of the architecture, the secure configuration and usage of the IC hardware platform and the IC Dedicated Support Software 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 by or even require support of the Security IC Embedded Software.
N7021 VA provides high security for smartcard applications and in particular for being used in the banking and finance market, in electronic commerce or in governmental applications. Hence, N7021 VA shall maintain

- the integrity and the confidentiality of code and data stored in its memories,
- the different TOE 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 N7021 VA basically provides a hardware platform for an implementation of a smart card application with

- functionality to calculate Data Encryption Standard (Triple-DES) with up to three keys,
- hardware to calculate 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,
- a Hybrid Deterministic Random Number Generator,
- a Hybrid Physical Random Number Generator,
- memory management control,
- cyclic redundancy check (CRC) calculation,
- ISO/IEC 7816 contact interface with UART,
- ISO/IEC14443A 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. The support for large integer arithmetic operations does not provide security functionality like cryptography. The Security IC Embedded Software that implements an asymmetric cryptographic algorithm is not included in this Security Target, but the support for large integer arithmetic operations is a security relevant component of the TOE, which resists to the attacks mentioned in this Security Target and operates correctly as specified in the data sheet. The same scope is applied to the CRC calculation. Similarly, even though single DES and two-key version of TDES are supported, they are not within the scope of evaluation.

1.3.2 TOE Type

The TOE NXP Secure Smart Card Controller N7021 VA is provided as IC hardware platform with IC Dedicated Software 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

N7021 VA is manufactured in $90nm$ CMOS technology. A block diagram of the IC hardware is depicted in Figure 1.1.

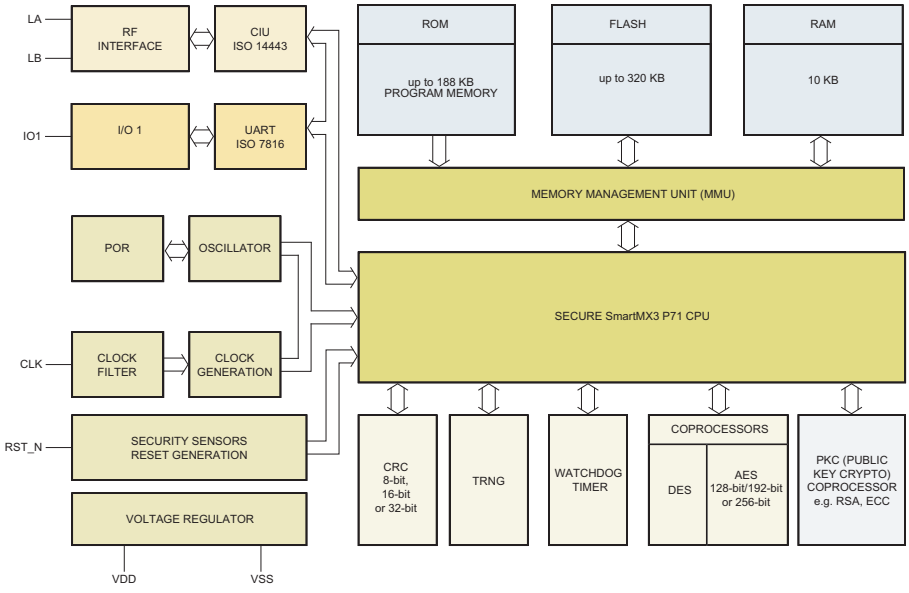

Fig. 1.1: Block Diagram

N7021 VA consists of the IC hardware and IC Dedicated Software. The IC Dedicated Software is composed of IC Dedicated Test Software for test purposes and IC Dedicated Support Software. The IC Dedicated Test Software contains the Test Software, the IC Dedicated Support Software is composed of the Boot Software, the Firmware Interface, the Shared OS Libraries, the Symmetric Crypto Library, the System Mode OS and the Flashloader OS. All other software is called Security IC Embedded Software. The Security IC Embedded Software is not part of the TOE. All components of the TOE are listed in section 1.4.1.1.

1.4.1.1 TOE components

Type *All configurations*	**Name**	**Version**	**Form of Delivery**
IC Hardware	N7021	VA	Wafer, modules and package
IC Dedicated Test Software	Test Software	20.0	On-chip software
IC Dedicated Support Software	Boot Software	20.0	On-chip software
	Firmware Interface	20.0	On-chip software
Document	SmartMX3 family P71D320 Overview, pinning and electrical characteristics, Product Data Sheet [25]	3.1	PDF via NXP DocStore
Document	SmartMX3 N7021 Instruction Set Manual, Product Data Sheet addendum [16]	1.4	PDF via NXP DocStore

Type	Name	Version	Form of Delivery
Document	SmartMX3 family N7021 Wafer and delivery specification, Data Sheet addendum [24]	1.3	PDF via NXP DocStore
Document	SmartMX3 N7021 Post Delivery Configuration Post Delivery Configuration, Data Sheet addendum [21]	1.1	PDF via NXP DocStore
Document	SmartMX3 N7021 Chip Health Mode, Data Sheet addendum [14]	1.0	PDF via NXP DocStore
Document	SmartMX3 N7021 Peripheral Configuration and Use, Data Sheet addendum [20]	1.4	PDF via NXP DocStore
Document	SmartMX3 N7021 MMU configuration & FW interface, Data Sheet addendum [18]	1.5	PDF via NXP DocStore
Document	SmartMX3 N7021 Inter-Card Communication, Data Sheet addendum [17]	1.1	PDF via NXP DocStore
Document	SmartMX3 N7021 NVM Operate Function, Data Sheet addendum [19]	1.0	PDF via NXP DocStore
Document	NXP Secure Smart Card Controller N7021 Information on Guidance and Operation, Guidance and Operation Manual [13]	1.4	PDF via NXP DocStore
Deliverables of the Flashloader OS			
IC Dedicated Support Software	Flashloader OS	20.0	On-chip software
Document	SmartMX3 N7021 FlashLoader, Product Data Sheet addendum [15]	1.3	PDF via NXP DocStore
Deliverables of the Shared OS Libraries			
IC Dedicated Support Software	Shared OS Libraries	20.0	On-chip software
Library File	libComm		SDK installer via NXP DocStore
Library File	libCrc		SDK installer via NXP DocStore
Library File	libMem		SDK installer via NXP DocStore
Library File	libFL		SDK installer via NXP DocStore
Document	SmartMX3 N7021 Shared OS libraries, Data Sheet addendum [22]	1.2	PDF via NXP DocStore
Deliverables of the System Mode OS			
IC Dedicated Support Software	System Mode OS	20.0	On-chip software

Type	Name	Version	Form of Delivery
Document	SmartMX3 N7021 NXP System Mode OS Interface, Data Sheet addendum [23]	1.6	PDF via NXP DocStore
Deliverables of the Symmetric Crypto Library			
IC Dedicated Support Software	Crypto Library Iron	2.0.6-01	On-chip software
Library Files	Crypto Library Iron	2.0.6-01	SDK installer via NXP DocStore
Document	Crypto Library V1.0 on N7021 VA, Symmetric Cipher Library (SymCfg), User manual [28]	1.2	PDF via NXP DocStore
Document	N7021 Crypto Library, RNG Library, User manual [27]	1.3	PDF via NXP DocStore
Document	N7021 Crypto Library, Utils Library, User manual [29]	1.1	PDF via NXP DocStore
Document	Crypto Library Iron on N7021 VA, Information on Guidance and Operation, Guidance and Operation Manual [7]	2.0	PDF via NXP DocStore

Tab. 1.1: Components of the TOE

The IC Hardware is delivered to the customer by secure transport in parcels sealed with special tape. The customer can examine the tape sealing to make sure that the TOE has not been manipulated during transport. The documentation can be downloaded by the customer from the NXP DocStore website after registration. Library files (object files, header files and linker scripts) are also made available to the customer via NXP DocStore, as part of a downloadable and installable SDK.
The logical dependencies of the TOE components are given in Section 1.4.3.2.
The IC Hardware is identified by a nameplate that is located in the layout of the chip (see [24] how to inspect the nameplate). The IC Dedicated Software is identified by 'IC Dedicated Software version', which can be read out by the Security IC Embedded Software via a GetVersion command as described in [14].

1.4.2 Evaluated Configurations

The N7021 VA can be delivered with various configuration options as described in this section. The configuration options are divided into two groups: major configuration options according to section 1.4.2.1 and minor configuration options according to section 1.4.2.2.

1.4.2.1 Major configuration options

Three major configurations can be chosen by the customer during the ordering process:

- Configuration based on 320 kBytes of Flash memory as code space
- Configuration based on 240 kBytes of Flash memory as code space
- Configuration based on 144 kBytes of ROM memory as code space

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.

1.4.2.2 Minor configuration options

Minor configurations are chosen by the customer during the ordering process as detailed in Table 1.2.

Product option	Choices	Description
Customer Type	System Mode Customer, **User Mode Customer**	Select the hierarchical interaction model of the Security IC Embedded Software.
Use Flash Loader	**Yes**, No	If selected, allow the download of an arbitrary card image into logical card B using the Flashloader.
UID Option	**7B UID**, 4B FNUID, 10B UID	Determines the UID setting.
Enable Contactless Interface	**Enabled**, Disabled	If enabled, the contactless interface is activated in the product configuration.
Input Capacitance	**17pF**, 56pF, 70pF	Nominal input capacitance for ISO/IEC 14443 contactless interface.
Data Rate	**All**, 106kbps, 106-848kbps, 106-848kbps and VHBR rates up to 3.2Mbps	Set the available data rates.
Enable Contact Interface	**Enabled**, Disabled	If enabled, the contact interface is activated in the product configuration.
PUF option	**Enabled**, Disabled	If enabled, the device supports the PUF security functionality.
MIFARE DESFire EV2 Option[1]	**Disabled**, 2K, 4K, 8K, 32K	Configure the available MIFARE DESFire image in logical card A of the system[1].
MIFARE Plus EV1 Option[1]	**Disabled**, 2K, 4K	Configure the available MIFARE PLUS2 image in logical card A of the system[1].
Enable Chip Health Mode	**Enabled**, Disabled	Enable the availability of Chip Health Mode (CHM).

Tab. 1.2: Evaluated minor configuration options

Regardless of the minor configuration options selected, the TOE will always remain in a certified configuration.

1.4.2.3 Post Delivery Configuration

Post Delivery Configuration (PDC) can be used by the customer after the TOE has been delivered by NXP. These options can be used to tailor the TOE to specific customer requirements. The Post Delivery Configuration settings 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.

[1] MIFARE emulations are not part of the TSF and are therefore not in scope of this certification.

Table 1.3 lists those configuration parameters that can be set by PDC in the NXP Secure Smart Card Controller N7021 VA.

PDC option	Description
Total requested Flash size	Define the total number of customer available Flash pages. PDC can only reduce this value.
Contact/contactless/dual operation mode	Define the operation mode which can be either "contact only", "contactless only", or "dual". Interfaces can only be deactivated by PDC if they were selected during ordering.
Disable cryptographic functions	Define the available cryptographic options. Each of the three functions (DES, AES, PKCC) can be independently disabled.
Outside Anti-Wear partition size Card B	Defines the outside anti-wear partition flash size available for logical card B.
Inside Anti-Wear partition size Card B and Free Page Pool (FPP) size	Defines the inside anti-wear partition flash page size of logical card B and the number of additional Free Page Pool pages. Wear-levelling increases Flash endurance.
Default logical card	Define which logical card SM (A or B) should be launched after finishing the boot sequence.
Default OS	Define which operating system (either OS 1 or OS 2) should be launched after finishing the boot sequence of the selected logical card.
Card A/Card B RAM partition split	Define how the RAM is split between Card A and Card B.
Basic control setting and codebase for OS1 in Card B	Set the codebase (memory offset) and options for OS1 in Card B.
Basic control setting and codebase for OS2 in Card B	Set the codebase (memory offset) and options for OS2 in Card B.

Tab. 1.3: Post Delivery Configuration

As indicated in the description of the PDC options, they can only be used to downgrade some configurations, it is not possible to enable functionality. The Post Delivery Configuration can be accessed using chip health mode functionality in combination with the ISO/IEC 7816 contact interface. PDC configures the availability of TSF. Deactivating TSF is identical to not utilizing the functionality and therefore the TOE will remain in a certified configuration. For further details regarding PDC refer to [21].

1.4.2.4 Dependencies on minor configuration and PDC

Depending on the minor configuration options chosen during the ordering process, and on the changes made using PDC, some of the security functionality mentioned in this ST is no longer available. Table 1.4 below lists these dependencies.

Option	Feature	SFRs comment
Use Flash Loader	SS.Loader	SFRs are still in place to ensure correct blocking of functionality.

Option	Feature	SFRs comment
Chip Health Mode	SS.RECONFIG	Feature CHM is not available, SS.RECONFIG itself is still available. PDC also available via System Mode API.
Disable DES	SS.HW_TDES, SS.SW_DES	Related SFRs are deactivated. (SF.Object_Reuse is still available)
Disable AES	SS.HW_AES, SS.SW_AES, SF.PUF, SS.RECONFIG	Related SFRs are deactivated. AES functionality is mandatory for SF.PUF. SS.RECONFIG needs AES for PDC configuration and CHM authentication. (SF.Object_Reuse is still available)
PUF option	SF.PUF	Related SFRs are deactivated.

Tab. 1.4: Dependencies on minor configuration and PDC

1.4.2.5 Evaluated package types

The commercial types are named according to the format *P7nxeeeypp(p)/mvrrff(o)*.
The characters in the above format are coded as described in Table 1.5 and Table 1.6. The commercial type name is composed of fixed symbols, which are detailed in Table 1.5, and variable entries, which are filled in according to the rules in Table 1.6.

Variable	Description	Values	Evaluated Options
n	indicates ROM or Flash product	numeric	'0' for ROM, '1' for Flash
x	Interface and Feature Configuration	alpha numeric	'D' for Dual Interface
eee	Indication of Flash or ROM Size, depending on variable *n*	alpha numeric	
y	MIFARE Emulation Option Configuration	alphanumeric	'P' for no MIFARE emulation, 'D' for DESFire EV2, 'M' for MIFARE Plus EV1
pp(p)	Package delivery type	alpha numeric	see table 1.6
/	separator (mandatory)		
m	Manufacturer identifier	alpha numeric	'9' for TSMC
v	Version of mask set	alphabetic	'A' for HW version VA
rr	NCN number, which identifies the NXP content at TOE delivery	alpha numeric	specific to a certain combination of major and minor options and IC Dedicated Software version and other NXP data elements
ff	CCN number, which identifies the customer content at TOE delivery	alpha numeric	specific to customer identity and content of code and data to be uploaded on behalf of the customer
(o)	Option	alpha numeric, optional	

Variable	Description	Values	Evaluated Options

Tab. 1.5: Variable Definitions for Commercial Type Names

Type	Description
U*nn*	Wafer not thinner than 50μm. The numbers "*nn*" in "U*nn*" identify a specific wafer delivery type (thickness, manufacturing process and packing option)
X*nn*	Chip card module. The numbers "*nn*" in "X*nn*" identify a specific chip module delivery type (module type, manufacturing process and packing option)
A*n*	Contactless chip card module or inlay type (assembly containing the TOE and a contactless antenna on a carrier material. The "*n*" in "A*n*" identifies a specific contactless module or inlay delivery type (type of contactless module or inlay, manufacturing process and packing option)

Tab. 1.6: Supported Package Types

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 during the ordering process, which is coupled to the NCN and CCN of the commercial type name.

1.4.3 Logical Scope of TOE

1.4.3.1 Hardware Description

The TOE distinguishes five TOE modes:

1. **Super System Mode (SSM)**
2. **Configuration Mode (CM)**
3. **Test Mode (Test Mode)**
4. **System Mode (SM)**
5. **User Mode (UM)**

The Super System Mode is not available to the Security IC Embedded Software. In Super System Mode the TOE executes the Boot Software resp. the IC Dedicated Test Software. Notice that the Firmware Interface also executes in Super System Mode and other parts are executed in System Mode and can be accessed via so-called system calls either from User Mode or System Mode. The Security IC Embedded Software may execute in System Mode or User Mode. Note also that the CPU itself only distinguishes between the User Mode, the System Mode and the Super System Mode. From CPU's perspective there is no difference between the Test Mode, the

Configuration Mode and the Super System Mode. The difference from system perspective is only that the Test Mode and Configuration Mode can extend their access rights to Special Function Registers compared to what is visible in Super System Mode (it can grant access to test features). However, this is enforced by the Memory Management Unit where the Test Mode and Configuration Mode are modelled as an own mode that has extended access rights compared to Super System Mode.
The N7021 VA is able to control two different logical phases. After production of the Security IC every start-up or reset completes with execution of the IC Dedicated Test Software. The test functionality is disabled at the end of the production test. Afterwards, every start-up or reset ends up in System Mode or User Mode, depending on the minor configuration 'Customer Type' selected by the customer.
System Mode and User Mode are available to the developer of the Security IC Embedded Software. System Mode has broader 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. Both are interrelated to the activities of the CPU, the Memory Management Unit, interrupt control, I/O configuration, NVM, timers and the coprocessors.
The N7021 VA provides interrupts. Interrupts force a jump to a specific fixed vector address in the ROM or Flash. Any interrupt can therefore be controlled and guided by a specific part of the Security IC Embedded Software. In addition, N7021 VA provides user calls and system calls. These calls have to be explicitly done by the Security IC Embedded Software via dedicated CPU instructions. A user call starts the execution of related code dedicated to one lower privileged mode (Super System Mode to System Mode or System Mode to User Mode). A system call starts the execution of related code dedicated to one higher privileged mode (User Mode to System Mode or System Mode to Super 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 N7021 VA incorporates 320 kBytes of Flash, 10 kBytes of RAM and up to 188 kBytes of program memory available in ROM. Access control to all three memory types is enforced by a Memory Management Unit (MMU). The System Mode OS provides a simplification of the resource management (e.g. MMU firewall settings, dynamic segment setup, peripheral access control). The MMU partitions each memory into several parts, defined as objects in the Hardware Access Control Policy (see section 6.1.6).
The Triple-DES coprocessor supports single DES and Triple-DES operations. Only Triple-DES in 3-key operation (168-bit) is in the scope of this evaluation. The AES coprocessor supports AES operation with three different key lengths of 128, 192 or 256 bit. The Public Key Crypto Coprocessor (PKCC) coprocessor supplies basic arithmetic functions to support implementation of asymmetric cryptographic algorithms by the Security IC Embedded Software. The random number generator provides true random numbers without pseudo random calculation. The deterministic random number generator provides pseudo-random calculation seeded by the true random number generator. The CRC coprocessor provides CRC generation polynomial CRC-8, CRC-16 and CRC-32.
The N7021 VA operates with a single external power supply of 1.8V, 3V or 5V nominal. Alternatively the TOE can be supplied via the RF interface by means of inductive coupling. The maximum external clock frequency used

for synchronization of the ISO/IEC 7816 communication is 10 MHz nominal, the CPU and all coprocessors are supplied exclusively with an internally generated clock signal which frequency can be selected by the Security IC Embedded Software. The N7021 VA 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 Flash such that data stored to these memories is encrypted. Chip shielding is added in form of active and passive shield over the whole chip surface. 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

Figure 1.2 illustrates the different pieces of software available on the TOE.

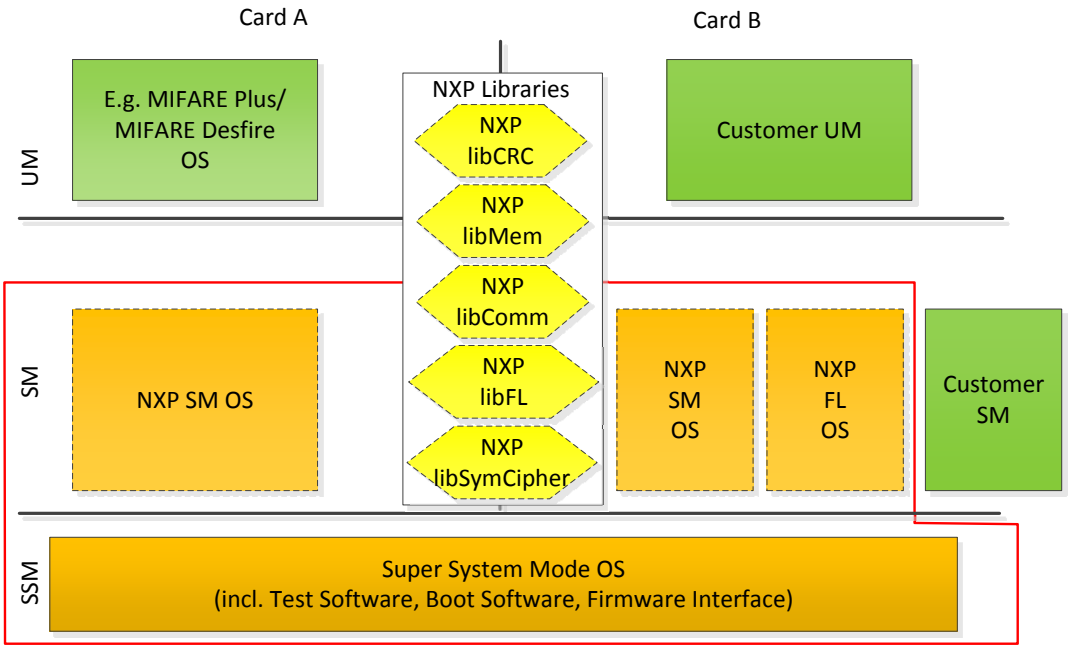

Fig. 1.2: Software Components of the TOE

The N7021 VA supports two logical cards (Card A and Card B). Both logical cards are divided into a User Mode and a System Mode. The logical location of the Security IC Embedded Software. depends on the usage of the IC hardware platform. Card A is reserved for Security IC Embedded Software developed by NXP (for example, a MIFARE Operating System). This code is stored in the UM-A_Code_Seg (which is the User Mode of Card A). Card B is available for Security IC Embedded Software developed by the customer, which can be stored in

either UM-B_Code_Seg (User Mode of Card B) or stored in the SM-B_Code_Seg (System Mode of Card B) if the customer is a System Mode customer. If a customer did not order any NXP developed Security IC Embedded Software product (such as MIFARE), then User Mode Card A is not present. The physical location of the Security IC Embedded Software can be either in ROM or in Flash and is not in the scope of this evaluation.
The separation between two logical cards (Card A and Card B) is provided by the so-called ”Vertical IP firewall” which allows having two completely separated logical cards on the same hardware without any unintended impact on each other. Because a logical card is also divided into a User Mode and a System Mode, it is possible to implement a security feature called ”Secure User Mode Box”.
The ”Secure User Mode Box” makes sure that NXP code and data residing in the User Mode of Card A (for example, a MIFARE Operating System) cannot have any security impact on the certified IC configuration. This is achieved because the System Mode OS in logical Card A restricts access of code executed in User Mode on logical Card A such that no influence to any other mode and card is possible. For the ”Secure User Mode Box” fixed values are NXP defined during production.
Using shared memory segments it is possible to share data or code between the separated logical cards. The owner of a memory block has to explicitly allow this kind of sharing. The libraries are shared between the logical cards using this mechanism, reducing the footprint, as code only has to be present on the device once. An Intercard communication mechanism allows the currently active card to send a message to the inactive card with a request for card switching. This mechanism allows for the handover of execution between the logical cards.
The IC Dedicated Test Software is developed by NXP and embedded in the Test Software. The 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 Flash’s manufacturer area and shutdown functions to ensure that security relevant test routines cannot be executed illegally after phase 3. This is stored in the SSM_Data_Seg. Moreover, the IC Dedicated Test Software is used to download patch code related to System Mode (stored in SM-A_Code_Seg or SM-B_Code_Seg) or User Mode (stored in UM-A_Code_Seg or UM-B_Code_Seg).
The IC Dedicated Support Software comprises the following parts:

1. The Boot Software is executed after each reset of the TOE, i.e. every time when the TOE starts. It sets up the TOE and does some basic configuration of the hardware based on the settings stored in SSM_Data_Seg, respectively. The Boot Software is stored in the SSM_Code_Seg.

2. The Firmware Interface is stored in the SSM_Code_Seg. It provides low-level flash management, the Post Delivery Configuration feature and basic system functionality like self-testing, error-counter handling and reset functionality. Notice, that Boot Software and IC Dedicated Test Software also access the Firmware Interface.

3. The System Mode OS is an Operating System developed by NXP and stored in the SM-A_Code_Seg and/or in the SM-B_Code_Seg and is accessed by the Security IC Embedded Software via system calls. It offers ready-to-use resource and access management for arbitrary customer applications that do not want to be exposed to the more low-level features such as MMU configuration. The System Mode OS on Card A is

mandatory when the Flashloader OS is part of the product, when library code is shared between the logical cards or when Card A is activated. The System Mode OS on Card B is mandatory for User Mode customers. For System Mode customers who do not need any NXP library and no Flashloader OS and no activated Card A, the System Mode OS is deactivated and cannot be executed.

4. The Shared OS Libraries are an optional module and can be stored in any Card and mode. It provides simplified communication, CRC and memory functions to the Security IC Embedded Software. The Shared OS Libraries are required by the Flashloader OS.

5. The Symmetric Crypto Library is an optional library which provides functions to access AES, DES, RNG and Secure Operations functionality to the Security IC Embedded Software. It is mandatory for the Flashloader OS. The physical location of the Symmetric Crypto Library depends on the configuration selected by the customer. For products that include the Flashloader OS, the Symmetric Crypto Library is located in ROM. For products without the Flashloader OS, the physical location depends on how the customer linked the Symmetric Crypto Library into their Security IC Embedded Software using the development tools.

6. The Flashloader OS is an optional module and stored in the SM-B_Code_Seg and cannot be directly accessed by the Security IC Embedded Software. It supports the download of code and data to Flash by the Composite Product Manufacturer before Operational Usage (e.g. during development). This functionality can be made unavailable after usage. When the Flashloader OS module is available, the Shared OS Libraries, Symmetric Crypto Library and System Mode OS become mandatory.

All logical dependencies of the IC Dedicated Support Software are described in the definitions above.

1.4.3.3 Security functionality dependency on optional software

As some of the modules and libraries are optional, the security functionality mentioned in this ST also becomes optional. If the Symmetric Crypto Library is not part of the TOE, the SFRs related to SS.SW_DES, SS.SW_AES, SS.SW_RNG and SF.Object_Reuse are not available. If the System Mode OS is optional, SS.Loader will no longer work (as System Mode OS functionally supports SS.Loader). SF.MEM_SUB will also not be available, as the System Mode OS makes sure that the MMU gets properly configured for this functionality.

1.4.3.4 Documentation

The documents containing a functional description and guidelines for the use of the security functionality, as needed to develop Security IC Embedded Software, are listed in Table 1.1. 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 [26]. 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 [26] the TOE supports the authentic delivery using the FabKey feature. For further details refer to the data sheet [25] and the guidance and operation manual [13].
During the design and the layout process only personnel involved in the specific development project for an IC have access to sensitive data. Different teams 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 NCN or CCN. After that step the wafers are completed with the product type specific data, including ROM and Flash Code, and data (if applicable) as identified by NCN and CCN. The test process of every die is performed by a test center of NXP. Delivery processes between the involved sites provide accountability and traceability of the TOE. NXP embeds the dice into modules, inlays or packages depending on the individual commercial type.
Information about non-functional items (so-called failed die) is stored on electronic media, and the access to this media made available with the delivery. The availability of major configuration options of the TOE in package types is detailed in section 1.4.2.1.

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 [26]. 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 N7021 VA can be used to perform various functions in a wide range of applications. Examples are identity cards, Banking Cards, Pay-TV, 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 high-end safeguarded applications, and is designed for embedding into chip cards according to ISO/IEC 7816 [9] and for contactless applications according to ISO/IEC 14443 [30]. 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 cannot be disabled by the Security IC Embedded Software in the following phases.

1.4.6 Interface of the TOE

The electrical interface of the N7021 VA are the pads to connect the lines power supply, ground, reset input, clock input, serial communication pad I/O1, as well as two pads (called LA and LB) for the antenna of the RF interface (See Figure 1.1). 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 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.

- Upon every start-up the Boot Software is executed in Super System Mode. This software initializes and configures the TOE. This comprises the selection of IC Dedicated Test Software (before TOE delivery) and of Security IC Embedded Software (after TOE delivery). Only in case the minor configuration option 'Enable Chip Health 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 Software provides no interface. In this case there is no possibility to interact with this software. The Boot Software is stored in the SSM_Code_Seg.
- Before TOE delivery the logical interface is defined by the IC Dedicated Test Software. This IC Dedicated Test Software is executed in Super System Mode and comprises the test operating system used for production testing. IC Dedicated Test Software is embedded in the Test 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 TOE 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 behavior the logical interface is defined by the Security IC Embedded Software.

2 Conformance Claims

This Security Target claims to be conformant to the Common Criteria version 3.1:

- Common Criteria for Information Technology Security Evaluation, Part 1 – Introduction and general model - Version 3.1 CCMB-2017-04-001, Revision 5, April 2017, [3]
- Common Criteria for Information Technology Security Evaluation, Part 2 – Security functional components, Version 3.1 CCMB-2017-04-002, Revision 5, April 2017, [4]
- Common Criteria for Information Technology Security Evaluation, Part 3 – Security Assurance Components, Version 3.1 CCMB-2017-04-003, Revision 5, April 2017, [5]

For the evaluation the following methodology will be used:

- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, Version 3.1 CCMB-2017-04-004, Revision 5, April 2017, [6]

This Security Target claims to be CC Part 2 extended and CC Part 3 conformant. The extended Security Functional Requirements are defined in chapter 5.

2.1 Package Claim

This Security Target claims conformance to the assurance package **EAL6 augmented**. The augmentations to EAL6 is ALC_FLR.1. In addition, the Security Target is augmented using the component ASE_TSS.2, which is chosen to include architectural information on the security functionality of the TOE.

Note: The Protection Profile (PP) ”Security IC Platform Protection Profile with Augmentation Packages” [26] to which this Security Target claims conformance (refer to section 2.2) 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.

2.2 PP Claim

This Security Target claims strict conformance to the Protection Profile (PP) "Security IC Platform Protection Profile with Augmentation Packages, Version 1.0, registered and certified by Bundesamt fuer Sicherheit in der Informationstechnik (BSI) under the reference BSI-PP-0084-2014" [26]. Thus, the concepts are used in the same sense. For the definition of terms refer to [26]. This chapter does not need any supplement in the Security Target. This conformance claim includes the following packages of security requirements out of those for Cryptographic Services defined in the Protection Profile [26]:

- Package ”TDES” (package conformant)
- Package ”AES” (package conformant)

This conformance claim includes the following packages of security requirements out of those for Loader defined in the Protection Profile [26]:

- Package ”Package 1: Loader dedicated for usage in Secured Environment only” (package conformant)
- Package ”Package 2: Loader dedicated for usage by authorized users only” (package conformant)

The TOE provides additional functionality, which is not covered in [26]. In accordance with Application Note 4 of [26] this additional functionality is added using the policy P.Add-Components (see section 3.3).

2.3 Conformance Claim Rationale

According to section 2.2 this ST claims strict conformance to the Security IC Platform Protection Profile with Augmentation Packages [26].
The TOE type defined in section 1.3.2 of this Security Target is a smartcard controller with IC Dedicated Software. This is consistent with the TOE definition for a Security IC in section 1.2.2 of [26].
The sections within this document where security problem definitions, security objectives and security requirements are defined, clearly state which of these items are taken from the Protection Profile and which are added in this Security Target. 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 are also clearly indicated.
The evaluation assurance level claimed for this TOE is shown in section 6.2 to include respectively exceed the requirements claimed by the PP (EAL4+).
These considerations show that the Security Target correctly claims conformance to the Security IC Platform Protection Profile with Augmentation Packages, [26].

3 Security Problem Definition

This chapter lists the assets, threats, assumptions and organizational security policies from the PP [26] and describes extensions to these elements in detail.

3.1 Description of Assets

All assets, which are defined in section 3.1 of the PP [26], are related to standard functionality. They are applied in this Security Target. These assets 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 provided by the TOE for the Security IC Embedded Software,
- Deficiency of random numbers.

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

- Logical design data, physical design data, IC Dedicated Software,
- Initialization data and pre-personalization data, Security IC Embedded Software, specific development aids, test and characterization related data, material for software development support, photo masks.

Note that the keys for cryptographic calculations using security services of the TOE are treated as User Data.

3.2 Threats

All threats, which are defined in section 3.2 of the PP [26], are valid for this Security Target.

T.Leak-Inherent **Inherent Information Leakage**
An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential user data as part of the assets.

T.Phys-Probing **Physical Probing**
An attacker may perform physical probing of the TOE in order

(i) to disclose user data while stored in protected memory areas,

(ii) to disclose/reconstruct the user data while processed or

(iii) to disclose other critical information about the operation of the TOE to enable attacks disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.

T.Malfunction **Malfunction due to Environmental Stress**

An attacker may cause a malfunction of TSF or of the Security IC Embedded Software by applying environmental stress in order to

(i) modify security services of the TOE or

(ii) modify functions of the Security IC Embedded Software

(iii) deactivate or affect security mechanisms of the TOE to enable attacks disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.

This may be achieved by operating the Security IC outside the normal operating conditions (Numbers 1, 2 and 9 in Figure 8 of the Security IC PP).

T.Phys-Manipulation **Physical Manipulation**

An attacker may physically modify the Security IC in order to

(i) modify user data of the Composite TOE,

(ii) modify the Security IC Embedded Software,

(iii) modify or deactivate security services of the TOE, or

(iv) modify security mechanisms of the TOE to enable attacks disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.

T.Leak-Forced **Forced Information Leakage**

An attacker may exploit information which is leaked from the TOE during usage of the Security IC in order to disclose confidential user data of the Composite TOE as part of the assets even if the information leakage is not inherent but caused by the attacker.

T.Abuse-Func **Abuse of Functionality**

An attacker may use functions of the TOE which may not be used after TOE Delivery in order to

(i) disclose or manipulate user data of the Composite TOE,

(ii) manipulate (explore, bypass, deactivate or change) security services of the TOE or

(iii) manipulate (explore, bypass, deactivate or change) functions of the Security IC Embedded Software or

(iv) enable an attack disclosing or manipulating the user data of the Composite TOE or the Security IC Embedded Software.

T.RND **Deficiency of Random Numbers**

An attacker may predict or obtain information about random numbers generated by the TOE security service for instance because of a lack of entropy of the random numbers provided.

The threats defined in the PP [26] are listed in Table 3.1.

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

Tab. 3.1: Threats defined in the Security IC Platform Protection Profile with Augmentation Packages

In compliance with Application Note 4 in the PP [26] the TOE provides additional functionality to protect against threats that appear when the TOE is used for multiple applications.
The TOE provides the Security IC Embedded Software running in System Mode with control of access to memories and hardware components by different applications running in User Mode. In this context, User Data of different applications is stored to such memory and processed by such hardware components. The Security IC Embedded Software controls all these User Data. Any access to User Data assigned to one application by another application contradicts separation between different applications and is considered as a threat.
The TOE shall avert the threats T.Unauthorised-Access and T.Secure-UM-Box-Border as specified below.

T.Unauthorised-Acce ss **Unauthorized Memory or Hardware Access**

Adverse action: An attacker may try to read, modify or execute code or data stored in restricted memory areas. 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 executed or data used in System Mode or User Mode may accidentally or deliberately access code or User Data of other applications. Any code executed or data used in System Mode or User Mode may accidentally or deliberately access hardware resources that are restricted to other applications.

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

Asset: Code executed by and data belonging to the IC Dedicated Support Software running in Super System Mode or Test Mode as well as code executed by and data belonging to the Security IC Embedded Software.

T.Secure-UM-Box-Bo rder **Secure User Mode Box Border**

Adverse action: An attacker may try to use malicious code placed in the Secure User Mode Box to modify the correct behavior of the IC Dedicated Software or the Security IC Embedded Software as well as read or modify code or data belonging to the Security IC Dedicated Software or the Security IC Embedded Software.

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

Asset: Code executed by and data belonging to the Security IC Dedicated Software as well as code executed by and data belonging to the Security IC Embedded Software.

Restrictions of access to memories and hardware resources, which are available to 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.
The threats defined in this Security Target are summarized in Table 3.2.

Name	Title
T.Unauthorised-Access	Unauthorized Memory or Hardware Access
T.Secure-UM-Box-Border	Secure User Mode Box Border

Tab. 3.2: Additional Threats defined in this ST

3.3 Organizational Security Policies

All security policies, which are defined in section 3.3 of the PP [26], are valid for this Security Target. Additionally the security policies P.Lim_Block_Loader (Package 1: Loader dedicated for usage in secured environment only), P.Ctlr_Loader (Package 2: Loader dedicated for usage by authorized users only) and P.Crypto-Service (Packages for Cryptographic Services) defined in the packages of the PP [26] apply also for this Security Target.

P.Process-TOE **Identification during TOE Development and Production**
An accurate identification must be established for the TOE. This requires that each instantiation of the TOE carries this unique identification.

P.Lim_Block_Loader **Limiting and Blocking the Loader Functionality**
The composite manufacturer uses the Loader for loading of Security IC Embedded Software, user data of the Composite Product or IC Dedicated Support Software in charge of the IC Manufacturer. He limits the capability and blocks the availability of the Loader in order to protect stored data from disclosure and manipulation.

P.Ctlr_Loader **Controlled usage to Loader Functionality**
Authorized user controls the usage of the Loader functionality in order to protect stored and loaded user data from disclosure and manipulation.

P.Crypto-Service **Cryptographic services of the TOE**
The TOE provides secure hardware based cryptographic services for the IC Embedded Software.

All security policies defined in the PP [26] are listed in Table 3.3.

Name	Title
P.Process-TOE	Identification during TOE Development and Production

Name	Title
P.Lim_Block_Loader	Limiting and Blocking the Loader Functionality
P.Ctlr_Loader	Controlled usage to Loader Functionality
P.Crypto-Service	Cryptographic services of the TOE

Tab. 3.3: Policies defined in the Security IC Platform Protection Profile with Augmentation Packages

In compliance with Application Note 5 in the PP [26], this Security Target defines one additional security policy as detailed below.
The TOE provides specific security functionality, which can be used by the Security IC Embedded Software. This specific security functionality is not derived from threats identified for the TOE. Instead, the Security IC Embedded Software decides how to use this security functionality to protect from threats for the composite product. Thus, security policy P.Add-Components is defined as follows.

P.Add-Components **Additional Specific Security Components**
The TOE shall provide the following additional security functionality to the Security IC Embedded Software:

- Self Testing
- A function to reset the device
- Integrity support of data stored to NVM
- Reconfiguration of customer selectable options according to Post Delivery Configuration
- PUF functionality
- Provide protection of residual information

The security policies defined in this Security Target are summarized in Table 3.4.

Name	Title
P.Add-Components	Additional Specific Security Components

Tab. 3.4: Additional Security Policies defined in this ST

3.4 Assumptions

All assumptions, which are defined in section 3.4 of the PP [26], are valid for this Security Target.

A.Process-Sec-IC **Protection during Packaging, Finishing and Personalisation**
It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the endconsumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).

A.Resp-Appl **Treatment of user data of the Composite TOE**
All user data of the Composite TOE are owned by Security IC Embedded Software. Therefore, it must be assumed that security relevant user data of the Composite TOE (especially cryptographic keys) are treated by the Security IC Embedded Software as defined for its specific application context.

The assumptions defined in the PP [26] are listed in Table 3.5.

Name	Title
A.Process-Sec-IC	Protection during Packaging, Finishing and Personalisation
A.Resp-Appl	Treatment of user data of the Composite TOE

Tab. 3.5: Assumptions defined in the Security IC Platform Protection Profile with Augmentation Packages

In compliance with Application Notes 6 and 7 in PP [26], this Security Target defines two additional assumptions as follows.

A.Check-Init **Check of initialization data by the Security IC Embedded Software**
The Security IC Embedded Software must provide a function to check initialization data. Such data is defined by the Composite Product Manufacturer and injected by the TOE Manufacturer into the non-volatile memory to provide the ability to identify and trace the TOE.

The following additional assumption considers specialized encryption hardware of the TOE.
The developer of the Security IC Embedded Software must ensure appropriate usage of key-dependent functions as defined below during phase 1 of the Security IC product life cycle [26].

A.Key-Function **Usage of Key-dependent Functions**
Key-dependent functions (if any) shall be implemented in the Security IC Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced).
Note that here the routines which may compromise keys when being executed are part of the Security IC Embedded Software. In contrast to this the threats T.Leak-Inherent and T.Leak-Forced address (i) the cryptographic routines which are part of the TOE and (ii) the processing of User Data including cryptographic keys.

The assumptions defined in this Security Target are summarized in Table 3.6.

Name	Title
A.Check-Init	Check of initialization data by the Security IC Embedded Software
A.Key-Function	Usage of Key-dependent Functions

Tab. 3.6: Additional Assumptions defined in this ST

4 Security Objectives

This chapter defines the security objectives that shall be met by the TOE, the Security IC Embedded Software Development Environment and the Operational Environment.

4.1 Security Objectives for the TOE

All security objectives for the TOE, which are defined in the PP [26], are applied to this Security Target. Additionally the security objectives O.Cap_Avail_Loader (Package 1: Loader dedicated for usage in secured environment only), O.Ctrl_Auth_Loader (Package 2: Loader dedicated for usage by authorized users only), O.TDES (Package TDES), O.AES (Package AES) defined in the packages of the PP [26] apply also for this Security Target.

O.Leak-Inherent **Protection against Inherent Information Leakage**

The TOE must provide protection against disclosure of confidential data stored and/or processed in the Security IC

- by measurement and analysis of the shape and amplitude of signals (for example on the power, clock, or I/O lines) and
- by measurement and analysis of the time between events found by measuring signals (for instance on the power, clock, or I/O lines).

This objective pertains to measurements with subsequent complex signal processing whereas O.Phys-Probing is about direct measurements on elements on the chip surface. Details correspond to an analysis of attack scenarios which is not given here.

O.Phys-Probing **Protection against Physical Probing**

The TOE must provide protection against disclosure/reconstruction of user data while stored in protected memory areas and processed or against the disclosure of other critical information about the operation of the TOE.
This includes protection against

- measuring through galvanic contacts which is direct physical probing on the chips surface except on pads being bonded(using standard tools for measuring voltage and current) or
- measuring not using galvanic contacts but other types of physical interaction between charges (using tools used in solid-state physics research and IC failure analysis)

with a prior reverse-engineering to understand the design and its properties and functions.
The TOE must be designed and fabricated so that it requires a high combination of complex equipment, knowledge, skill, and time to be able to derive detailed design information or other information which could be used to compromise security through such a physical attack.

O.Malfunction **Protection against Malfunctions**

The TOE must ensure its correct operation.
The TOE must indicate or prevent its operation outside the normal operating conditions where

reliability and secure operation has not been proven or tested. This is to prevent malfunctions. Examples of environmental conditions are voltage, clock frequency, temperature, or external energy fields.

O.Phys-Manipulation **Protection against Physical Manipulation**

The TOE must provide protection against manipulation of the TOE (including its software and TSF data), the Security IC Embedded Software and the user data of the Composite TOE.
This includes protection against

- reverse-engineering (understanding the design and its properties and functions),
- manipulation of the hardware and any data, as well as
- undetected manipulation of memory contents.

O.Leak-Forced **Protection against Forced Information Leakage**

The Security IC must be protected against disclosure of confidential data processed in the Security IC (using methods as described under O.Leak-Inherent) even if the information leakage is not inherent but caused by the attacker

- by forcing a malfunction (refer to ”Protection against Malfunctions (O.Malfunction)”) and/or
- by a physical manipulation (refer to ”Protection against Physical Manipulation (O.Phys-Manipulation)”).

If this is not the case, signals which normally do not contain significant information about secrets could become an information channel for a leakage attack.

O.Abuse-Func **Protection against Abuse of Functionality**

The TOE must prevent that functions of the TOE which may not be used after TOE Delivery can be abused in order to

(i) disclose critical user data of the Composite TOE,

(ii) manipulate critical user data of the Composite TOE,

(iii) manipulate Security IC Embedded Software or

(iv) bypass, deactivate, change or explore security features or security services of the TOE.

Details depend, for instance, on the capabilities of the Test Features provided by the IC Dedicated Test Software which are not specified here.

O.Identification **TOE Identification**

The TOE must provide means to store Initialisation Data and Pre-personalisation Data in its non-volatile memory. The Initialisation Data (or parts of them) are used for TOE identification.

O.RND **Random Numbers**
The TOE will ensure the cryptographic quality of random number generation. For instance random numbers shall not be predictable and shall have a sufficient entropy. The TOE will ensure that no information about the produced random numbers is available to an attacker since they might be used for instance to generate cryptographic keys.

O.Cap_Avail_Loader **Capability and availability of the Loader**
The TSF provides limited capability of the Loader functionality and irreversible termination of the Loader in order to protect stored user data from disclosure and manipulation.

O.Ctrl_Auth_Loader **Access control and authenticity for the Loader**
The TSF provides trusted communication channel with authorized user, supports confidentiality protection and authentication of the user data to be loaded and access control for usage of the Loader functionality.

O.TDES **Cryptographic service Triple-DES**
The TOE provides secure hardware based cryptographic services implementing the Triple-DES for encryption and decryption.

O.AES **Cryptographic service AES**
The TOE provides secure hardware based cryptographic services implementing the AES for encryption and decryption.

The security objectives of the TOE defined in the PP [26] are listed in Table 4.1.

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
O.Cap_Avail_Loader	Capability and availability of the Loader
O.Ctrl_Auth_Loader	Access control and authenticity for the Loader
O.TDES	Cryptographic service Triple-DES
O.AES	Cryptographic service AES

Tab. 4.1: Security Objectives of the TOE defined in the Security IC Platform Protection Profile with Augmentation Packages

In compliance with Application Notes 8 and 9 in the PP [26], additional security objectives for the TOE are defined below based on additional functionality provided by the TOE.

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.

O.NVM_INTEGRITY **Integrity Support of data stored to NVM**
The TOE shall provide detection and correction of failures in NVM memories to support integrity of contents stored there.

O.MEM_ACCESS **Area based Memory Access Control**
The TOE shall control access of CPU instructions to memory areas depending on memory partitioning and based on TOE modes Super System Mode, System Mode and User Mode. In Super System Mode, System Mode and User Mode the TOE shall control access also based on configuration. In User Mode, the TOE shall control access also based on memory segments, which are configured in System Mode when implementing a memory management scheme. This control shall be individual to each memory segment and consider different access rights.

O.SFR_ACCESS **Special Function Register Access Control**
The TOE shall control access of CPU instructions to Special Function Registers depending on the purpose of the register and based on TOE modes. The TOE shall provide System Mode with the ability to configure access rights for User Mode to Special Function Registers that interface to hardware components.

O.REUSE **Application reuse of Memory**
The TOE shall include measures to ensure that the memory resources being used by an application of the TOE cannot be disclosed to subsequent users of the same memory resource of another application.

O.Self-Test **Self Test**
The TOE shall include functionality to perform a self-test to detect physical manipulation.

O.PUF **Sealing/Unsealing user data**
The TOE shall provide a PUF functionality that supports sealing/unsealing of user data. Using this functionality, the user data can be sealed within the TOE and can be unsealed by the same TOE that the user data was sealed on. The PUF functionality comprises import/export of data, encryption/decryption of data and calculation of a MAC as a PUF authentication value.
Note: The PUF functionality provided by the TOE shall only be active if explicitly configured by the Security IC Embedded Software.

O.Reset **Reset function**
The TOE shall provide the Security IC Embedded Software with a function to reset the device.

O.Secure-UM-Box-FW **Secure User Mode Box Firewall**
The TOE shall provide separation between the Secure UM Box code and other parts of the TOE. The separation shall comprise software execution and data access.

The objectives of the TOE defined in this Security Target are summarized in Table 4.2.

Name	Title
O.CUST_RECONFIG	Post Delivery Configuration
O.NVM_INTEGRITY	Integrity Support of data stored to NVM
O.MEM_ACCESS	Area based Memory Access Control
O.SFR_ACCESS	Special Function Register Access Control
O.REUSE	Application reuse of Memory
O.Self-Test	Self Test
O.PUF	Sealing/Unsealing user data
O.Reset	Reset function
O.Secure-UM-Box-FW	Secure User Mode Box Firewall

Tab. 4.2: Security Objectives of the TOE defined in this ST

4.2 Security Objectives for the Security IC Embedded Software Development Environment

All security objectives for the Security IC Embedded Software development Environment, which are defined in the PP [26], are applied to this Security Target.

OE.Resp-Appl **Treatment of User Data**
Security relevant user data of the Composite TOE (especially cryptographic keys) are treated by the Security IC Embedded Software as required by the security needs of the specific application context.

The security objectives defined in the PP [26] are listed in Table 4.3.

Name	Title
OE.Resp-Appl	Treatment of User Data

Tab. 4.3: Security Objectives of the Development Environment defined in the Security IC Platform Protection Profile with Augmentation Packages

Clarification related to "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. If keys are imported into the TOE and/or derived from other keys, quality and confidentiality must be maintained. This implies that appropriate key management has to be realized in the environment.
In case the Security IC Embedded Software operates multiple applications on the TOE, OE.Resp-Appl must also be met. The Security IC Embedded Software must not disclose security relevant User Data of one application to another application when processed in or stored to the TOE.

4.3 Security Objectives for the Operational Environment

All security objectives for the operational environment, which are defined in the PP [26], are applied to this Security Target. Additionally the security objectives for the TOE environment OE.Lim_Block_Loader (Package 1: Loader dedicated for usage in secured environment only) and OE.Loader_Usage (Package 2: Loader dedicated for usage by authorized users only) defined in the packages of the PP [26] apply also for this Security Target.

OE.Process-Sec-IC **Protection during composite product manufacturing**
Security procedures shall be used after TOE Delivery up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use). This means that Phases after TOE Delivery up to the end of Phase 6 (refer to Section 1.2.3 of the Security IC PP) must be protected appropriately. For a preliminary list of assets to be protected refer to paragraph 96 of the Security IC PP.

OE.Lim_Block_Loader **Limitation of capability and blocking the Loader**
The Composite Product Manufacturer will protect the Loader functionality against misuse, limit the capability of the Loader and terminate irreversibly the Loader after intended usage of the Loader.

OE.Loader_Usage **Secure communication and usage of the Loader**
The authorized user must support the trusted communication channel with the TOE by confidentiality protection and authenticity proof of the data to be loaded and fulfilling the access conditions required by the Loader.

The security objectives defined in the PP [26] are listed in Table 4.4.

Name	Title
OE.Process-Sec-IC	Protection during composite product manufacturing
OE.Lim_Block_Loader	Limitation of capability and blocking the Loader
OE.Loader_Usage	Secure communication and usage of the Loader

Tab. 4.4: Security Objectives of the Operational Environment defined in the Security IC Platform Protection Profile with Augmentation Packages

The following additional security objectives for the operational environment are defined in this Security Target. The following security objective for the operational environment derives from assumption A.Check-Init. The TOE provides specific functionality that requires the TOE Manufacturer to implement measures for unique identification of the TOE. Security objective OE.Check-Init is defined to allow for such a TOE specific implementation.

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

The objectives for the operational environment defined in this Security Target are summarized in Table 4.5.

Name	Title
OE.Check-Init	Check of initialization data by the Security IC Embedded Software

Tab. 4.5: Security Objectives of the Operational Environment defined in this ST

4.4 Security Objectives Rationale

Section 4.4 in the PP [26] provides a rationale how the threats, organisational security policies and assumptions are addressed by the security objectives defined in the PP [26]. Table 4.6 summarizes how threats, organisational security policies and assumptions of the PP are addressed by security objectives defined in the PP and ST, respectively. All these items are in line with those in the PP [26].

Security Problem Definition	Security Objective	Notes
T.Leak-Inherent	O.Leak-Inherent	
T.Phys-Probing	O.Phys-Probing	
T.Malfunction	O.Malfunction O.Self-Test	
T.Phys-Manipulation	O.Phys-Manipulation O.Self-Test	
T.Leak-Forced	O.Leak-Forced	
T.Abuse-Func	O.Abuse-Func	
T.RND	O.RND	
P.Process-TOE	O.Identification	Phases 2–3
A.Process-Sec-IC	OE.Process-Sec-IC	Phases 4–6
A.Resp-Appl	OE.Resp-Appl	Phase 1
P.Lim_Block_Loader	O.Cap_Avail_Loader OE.Lim_Block_Loader	
P.Ctlr_Loader	O.Ctrl_Auth_Loader OE.Loader_Usage	

Security Problem Definition	Security Objective	Notes
P.Crypto-Service	O.TDES O.AES	

Tab. 4.6: Security Objectives (PP and ST) vs. Security Problem Definition (PP)

Table 4.7 summarizes how threats, organisational security policies and assumptions of this ST are addressed by security objectives defined in the PP and ST, respectively.

Security Problem Definition	Security Objective	Notes
T.Unauthorised-Access	O.MEM_ACCESS O.SFR_ACCESS	
T.Secure-UM-Box-Border	O.Secure-UM-Box-FW	
P.Add-Components	O.Self-Test O.Reset O.CUST_RECONFIG O.NVM_INTEGRITY O.PUF O.REUSE	
A.Check-Init	OE.Check-Init	Phases 1 and 4–6
A.Key-Function	OE.Resp-Appl	Phase 1

Tab. 4.7: Security Objectives (PP and ST) vs. Security Problem Definition (ST)

The rationale for additional mappings between Threats defined in the PP [26] and Security Objectives defined in this Security Target is given below.

Justification related to T.Malfunction:

Objective	Rationale
O.Malfunction	It is clear from the description of the objective, that the corresponding threat is removed if the objective is valid. More specifically, in every case the ability to use the attack method successfully is countered, if the objective holds.
O.Self-Test	This objectives requires that the TOE provides self-testing features for security critical components, thus contributing to cover this threat.

Justification related to T.Phys-Manipulation:

Objective	Rationale
O.Phys-Manipulation	It is clear from the description of the objective, that the corresponding threat is removed if the objective is valid. More specifically, in every case the ability to use the attack method successfully is countered, if the objective holds.
O.Self-Test	This objectives requires that the TOE provides self-testing features for security critical components, thus contributing to cover this threat.

The rationale for all items defined in this Security Target is given below.

Justification related to T.Unauthorised-Access:

Objective	Rationale
O.MEM_ACCESS	TOE must enforce memory partitioning with address mapping and control of access to memories in System Mode and User Mode. Access rights in User Mode must be explicitly granted by Security IC Embedded Software running in System Mode. Thus, security violations caused by accidental or deliberate access to restricted data, code and shared hardware resources can be prevented.
O.SFR_ACCESS	The TOE must enforce control of access to Special Function Registers in System Mode and User Mode. Access rights in User Mode must be explicitly granted by code running in System Mode. Thus, security violations caused by accidental or deliberate access to restricted data, code and shared hardware resources can be prevented.

Justification related to T.Secure-UM-Box-Border:

Objective	Rationale
O.Secure-UM-Box-FW	The objective addresses the threat directly by ensuring that code running inside the Secure User Mode Box and data belonging to the Secure User Mode Box is separated from the other parts of the TOE. Due to the separation the code running in the Secure User Mode Box cannot harm the code or data outside the Secure User Mode Box.

Justification related to P.Add-Components:

Objective	Rationale
O.Self-Test	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.Reset	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.CUST_RECONFIG	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.NVM_INTEGRITY	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.PUF	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.
O.REUSE	This objective covers the security policy because it requires the TOE to partly implement the functionality as required by the security policy.

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.

Justification related to A.Check-Init:

Objective	Rationale
OE.Check-Init	This objective requires the Security IC Embedded Software developer to implement a function as stated in this assumption.

Justification related to A.Key-Function:

Objective	Rationale
OE.Resp-Appl	The definition of this objective of the PP [26] is further clarified in this Security Target: 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 this objective.

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

5 Extended Components Definitions

This Security Target does not define extended components.

6 Security Requirements

This chapter defines the security requirements that shall be met by the TOE. These security requirements are composed of the security functional requirements and the security assurance requirements that the TOE must meet in order to achieve its security objectives. CC allows several operations to be performed on security requirements (on the component level); refinement, selection, assignment, and iteration are defined in section 8.1 of CC Part 1 [3]. These operations are used in the PP [26] and in this Security Target, respectively.

The **refinement** operation is used to add details to requirements, and thus, further intensifies a requirement.

The **selection** operation is used to select one or more options provided by the PP [26] 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 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 [26] contains an operation that is left uncompleted, the Security Target has to complete that operation.

6.1 Security Functional Requirements

All Security Functional Requirements (SFRs) of the TOE are presented in the following sections to support a better understanding of the combination of the PP [26] and this Security Target. Tables 6.1 and 6.2 summarize the SFRs defined in the PP and ST, respectively.

Name	Title
FAU_SAS.1[HW]	Audit Storage
FCS_COP.1[TDES_HW]	Cryptographic operation - TDES - Hardware Support
FCS_COP.1[TDES_SW]	Cryptographic operation - TDES - Software Support
FCS_COP.1[AES_HW]	Cryptographic operation - AES - Hardware Support
FCS_COP.1[AES_SW]	Cryptographic operation - AES - Software Support
FCS_CKM.4[TDES_SW]	Cryptographic key destruction - TDES - Software
FCS_CKM.4[AES_SW]	Cryptographic key destruction - AES - Software
FCS_RNG.1[HW]	Random Number Generation (Class PTG.2)
FCS_RNG.1[HDT]	Random Number Generation (Hybrid-Deterministic)
FCS_RNG.1[HPH]	Random Number Generation (Hybrid-Physical)
FDP_ACC.1[Loader]	Subset access control - Loader
FDP_ACF.1[Loader]	Security attribute based access control - Loader
FDP_ITT.1[HW]	Basic Internal Transfer Protection

Name	Title
FDP_IFC.1	Subset Information Flow Control
FDP_UCT.1	Basic data exchange confidentiality
FDP_UIT.1	Data exchange integrity
FDP_SDC.1[HW]	Stored data confidentiality
FDP_SDI.2[HW]	Stored data integrity monitoring and action
FMT_LIM.1[HW]	Limited Capabilities
FMT_LIM.1[Loader]	Limited Capabilities
FMT_LIM.2[HW]	Limited Availability
FMT_LIM.2[Loader]	Limited Availability
FPT_FLS.1	Failure with Preservation of Secure State
FPT_ITT.1[HW]	Basic Internal TSF Data Transfer Protection
FPT_PHP.3	Resistance to Physical Attack
FRU_FLT.2	Limited Fault Tolerance
FTP_ITC.1	Inter-TSF trusted channel

Tab. 6.1: Security Functional Requirements defined in the Security IC Platform Protection Profile with Augmentation Packages

Name	Title
FCS_COP.1[AES_PUF]	Cryptographic operation - PUF based AES
FCS_COP.1[MAC_PUF]	Cryptographic operation - PUF based MAC
FCS_CKM.1[PUF]	Cryptographic Key Generation - PUF
FCS_CKM.4[PUF]	Cryptographic Key Destruction - PUF
FDP_ACC.1[MEM]	Subset Access Control (Memories)
FDP_ACC.1[SFR]	Subset Access Control (Special Function Registers)
FDP_ACC.1[SUB]	Subset Access Control (Secure User Mode Box)
FDP_ACF.1[MEM]	Security Attribute Based Access Control (Memories)
FDP_ACF.1[SFR]	Security Attribute Based Access Control (Special Function Registers)
FDP_ACF.1[SUB]	Security Attribute Based Access Control (Secure User Mode Box)
FDP_RIP.1[SW]	Subset Residual Information Protection
FMT_MSA.1[MEM]	Management of Security Attributes (Memories)
FMT_MSA.1[SFR]	Management of Security Attributes (Special Function Registers)
FMT_MSA.1[SUB]	Management of Security Attributes (Secure User Mode Box)
FMT_MSA.3[MEM]	Static Attribute Initialization (Memories)
FMT_MSA.3[SFR]	Static Attribute Initialization (Special Function Registers)
FMT_MSA.3[SUB]	Static Attribute Initialization (Secure User Mode Box)

Name	Title
FMT_SMF.1[HW]	Specification of Management Functions (Hardware)
FMT_SMF.1[SW]	Specification of Management Functions (Software)
FPT_TST.1	TSF Testing

Tab. 6.2: Security Functional Requirements defined in this ST

6.1.1 SFRs of the Protection Profile

All SFRs, which are defined in the PP [26] as well as those taken from the augmentation packages from the PP, are summarized in Table 6.1. Some of these SFRs are defined in CC Part 2 [4] and eventually subject to refinement, selection, assignment and/or iteration operation in the PP [26]. Others are newly defined in the PP [26].

SFRs FDP_ITT.1 and FPT_ITT.1 are defined in CC Part 2 [4] and are subject to refinement, selection and assignment operations in the PP [26]. The selection operations are further extended in this Security Target, which results in the following SFRs. Iteration [HW] is done here to prepare for other iterations that address any future major configurations of the TOE. The TOE shall meet requirement 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* to prevent the *disclosure* 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 co-processor) are seen as physically-separated parts of the TOE.

The TOE shall meet requirement 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* 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 co-processor) are seen as separated parts of the TOE.

The SFR FAU_SAS.1 is defined in the PP [26] and there is subject to two assignment operations. A third assignment operation is left open in the PP [26]. This operation assigns the type of persistent memory to which audit information is stored, and is filled in by this Security Target. In addition, the operation, which assigns the list

of audit information, is further extended in this Security Target. Iteration [HW] is done here to prepare for other iterations that address any future major configurations of the TOE. This results in the following SFR:

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* with the capability to store *the Initialisation Data, Pre-personalisation Data and customer-specific Data* in the *Flash*.

For FCS_RNG.1.1 the PP [26] 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 [26] 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 [26] have been replaced by operations defined in chapter 3 of [1] and the open operations of the partially filled in operations in the statement of the security requirements in section 4.4 of [1] for better readability. Note that the selection operation for the RNG type has already been filled in by the PP [26]. Iteration [HW] is done here to differentiate the random number hardware support from the random number software support. This results in the following SFR:

FCS_RNG.1[HW]	**Random Number Generation (Class PTG.2)**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FCS_RNG.1.1[HW]	The TSF shall provide a *physical* 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 that have been generated after the total failure of the entropy source.*

(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*. 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.

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

(PTG.2.6) Test procedure A [1] 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 definition of the Security Functional Requirement FCS_RNG.1 has been taken from [1].

Note: The functional requirement FCS_RNG.1[HW] is a refinement of FCS_RNG.1 defined in PP [26] according to [1].

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

Note: Application Note 20 in [26] requires that the Security Target specifies for the security capabilities in FCS_RNG.1.1[HW] how the results of the total failure test of the random source are provided to 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. 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 the term "bit" means measure of the Shannon-Entropy. The value "7.976" is assigned due to the requirements of "AIS31", [2].

By this, all assignment/selection operations are performed for FCS_RNG.1. This Security Target does not perform any other/further operations than stated in [1].
In addition to FCS_RNG.1[HW] the Symmetric Crypto Library provides a hybrid deterministic and hybrid physical random number generator:

FCS_RNG.1[HDT] **Random Number Generation (Hybrid-Deterministic)**

Hierarchical-To No other components.

Dependencies No dependencies.

FCS_RNG.1.1[HDT] The TSF shall provide a *hybrid deterministic* random number generator that implements:

(DRG.4.1) *The internal state of the RNG shall use PTRNG of class PTG.2 (as defined in [1]) as random source.*

(DRG.4.2) *The RNG provides forward secrecy (as defined in [1]).*

(DRG.4.3) *The RNG provides backward secrecy even if the current internal state is known (as defined in [1]).*

(DRG.4.4) *The RNG provides enhanced forward secrecy on demand (as defined in [1]).*

(DRG.4.5) *The internal state of the RNG is seeded by an PTRNG of class PTG.2 (as defined in [1]).*

FCS_RNG.1.2[HDT] The TSF shall provide *random numbers* that meet:

(DRG.4.6) *The RNG generates output for which 2^{48} strings of bit length 128 are mutually different with probability at least $1 - 2^{-24}$.*

[1] Note: according par.295 in [2] the assignment may be empty.

(DRG.4.7) *Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A (as defined in [1]).*

Note: The Crypto Library Software provides the Security IC Embedded Software with separate functionality to initialise the Crypto Library Software random number generator (which includes the chi-square test of the Hardware true random number generator) and to generate random data. It is the responsibility of the user to trigger the initialisation of the Crypto Library Software RNG before generating random data. The Crypto Library Software RNG will automatically trigger a reseed required by SP800-80A. If the use case of the user requires more frequent reseeding, then the user is responsible to trigger the reseed of the software RNG. Therefore, the user may use the software RNG reseed functionality or configure the RNG to enable the prediction resistance option.

Note: The Crypto Library does not prevent the operating system from accessing the hardware RNG. If the hardware RNG is used by the operating system directly, it has to be decided based on the Security IC Embedded Software security needs, what kind of tests has to be performed and what requirements wil have to be applied for this test. In this case the developer of the Security IC Embedded Software must ensure that the conditions prescribed in the user guidance manual are met.

Note: Only if the chi-square test succeeds the hardware RNG seeds the software RNG implemented as part of the Crypto Library Software.

FCS_RNG.1[HPH]	**Random Number Generation (Hybrid-Physical)**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FCS_RNG.1.1[HPH]	The TSF shall provide a *hybrid physical* random number generator that implements:

(PTG.3.1) *A total failure test detects a total failure of entropy source immediately when the RNG has started. When a total failure has been detected no random numbers will be output.*

(PTG.3.2) *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 that have been generated after the total failure of the entropy source.*

(PTG.3.3) *The online test shall detect non-tolerable statistical defects of the raw random number sequence (i) immediately when the RNG is started, and (ii) while the RNG is being operated. The TSF must not output any random numbers before the power-up online test and the seeding of the DRG.3 postprocessing algorithm have been finished successfully or when a defect has been detected.*

(PTG.3.4) *The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.*

(PTG.3.5) *The online test procedure checks the raw random number sequence. It is triggered continuously. The online test is suitable for detecting nontolerable statistical defects of the statistical properties of the raw random numbers within an acceptable period of time.*

(PTG.3.6) *The algorithmic post-processing algorithm belongs to Class DRG.3 with cryptographic state transition function and cryptographic output function, and the output data rate of the post-processing algorithm shall not exceed its input data rate.*

FCS_RNG.1.2[HPH] The TSF shall provide *random numbers* that meet:

(PTG.3.7) *Statistical test suites cannot practically distinguish the random numbers from output sequences of an ideal RNG. The random numbers must pass test procedure A (as defined in [1]).*

(PTG.3.8) *The internal random numbers shall use PTRNG of class PTG.2 as random source for the post-processing.*

FDP_IFC.1	**Subset Information Flow Control**
Hierarchical-To	No other components.
Dependencies	FDP_IFF.1 Simple security attributes
FDP_IFC.1.1	The TSF shall enforce the *Data Processing Policy* on *all confidential data when they are processed or transferred by the TOE or by the Security IC Embedded Software*.

The following Security Function Policy (SFP) *Data Processing Policy* is defined for the requirement ”Subset information flow control (FDP_IFC.1)”: User data of the Composite TOE and TSF data shall not be accessible from the TOE except when the Security IC Embedded Software decides to communicate the user data of the Composite TOE via an external interface. The protection shall be applied to confidential data only but without the distinction of attributes controlled by the Security IC Embedded Software.

FMT_LIM.1[HW]	**Limited Capabilities**
Hierarchical-To	No other components.
Dependencies	FMT_LIM.2 Limited availability.
FMT_LIM.1.1[HW]	The TSF shall be designed and implemented in a manner that limits their capabilities so that in conjunction with ”Limited availability (FMT_LIM.2)” the following policy is enforced: *Deploying Test Features after TOE Delivery does not allow user data of the Composite TOE to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks*.

FMT_LIM.2[HW]	**Limited Availability**
Hierarchical-To	No other components.
Dependencies	FMT_LIM.1 Limited capabilities.

FMT_LIM.2.1[HW]	The TSF shall be designed and implemented in a manner that limits their availability so that in conjunction with "Limited capabilities (FMT_LIM.1)" the following policy is enforced: *Deploying Test Features after TOE Delivery does not allow user data of the Composite TOE to be disclosed or manipulated, TSF data to be disclosed or manipulated, software to be reconstructed and no substantial information about construction of TSF to be gathered which may enable other attacks*.
FDP_SDC.1[HW]	**Stored data confidentiality**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FDP_SDC.1.1[HW]	The TSF shall ensure the confidentiality of the information of the user data while it is stored in the *RAM and Non-Volatile Memory*.
FDP_SDI.2[HW]	**Stored data integrity monitoring and action**
Hierarchical-To	FDP_SDI.1 Stored data integrity monitoring
Dependencies	No dependencies.
FDP_SDI.2.1[HW]	The TSF shall monitor user data stored in containers controlled by the TSF for *modification, deletion, repetition or loss of data* on all objects, based on the following attributes: *integrity check information associated with the data stored in memories*.
FDP_SDI.2.2[HW]	Upon detection of a data integrity error, the TSF shall *perform an error correction if possible and a Security Reset if not*.
FPT_FLS.1	**Failure with Preservation of Secure State**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: *exposure to operating conditions which may not be tolerated according to the requirement Limited fault tolerance (FRU_FLT.2) and where therefore a malfunction could occur*.
Refinement:	The term "failure" above also covers "circumstances". The TOE prevents failures for the "circumstances" defined above.
FPT_PHP.3	**Resistance to Physical Attack**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FPT_PHP.3.1	The TSF shall resist *physical manipulation and physical probing* to the *TSF* by responding automatically such that the SFRs are always enforced.

Refinement: The TSF will implement appropriate mechanisms to continuously counter physical manipulation and physical probing. Due to the nature of these attacks (especially manipulation) the TSF can by no means detect attacks on all of its elements. Therefore, permanent protection against these attacks is required ensuring that security functional requirements are enforced. Hence, ”automatic response” means here

(i) *assuming that there might be an attack at any time and*

(ii) *countermeasures are provided at any time.*

FRU_FLT.2	**Limited Fault Tolerance**
Hierarchical-To	FRU_FLT.1 Degraded fault tolerance
Dependencies	FPT_FLS.1 Failure with preservation of secure state.
FRU_FLT.2.1	The TSF shall ensure the operation of all the TOE’s capabilities when the following failures occur: *exposure to operating conditions which are not detected according to the requirement Failure with preservation of secure state (FPT_FLS.1).*

Refinement: The term ”failure” above means ”circumstances”. The TOE prevents failures for the ”circumstances” defined above.

6.1.1.1 SFRs for Augmentation Package “Loader Package1”

FMT_LIM.1[Loader]	**Limited Capabilities**
Hierarchical-To	No other components.
Dependencies	FMT_LIM.2 Limited availability.
FMT_LIM.1.1[Loader]	The TSF shall be designed and implemented in a manner that limits its capabilities so that in conjunction with ”Limited availability (FMT_LIM.2[Loader])” the following policy is enforced: *Deploying Loader functionality after switching to LifeCycleState.Release does not allow stored user data to be disclosed or manipulated by unauthorized user.*

FMT_LIM.2[Loader]	**Limited Availability**
Hierarchical-To	No other components.
Dependencies	FMT_LIM.1 Limited capabilities.
FMT_LIM.2.1[Loader]	The TSF shall be designed in a manner that limits its availability so that in conjunction with ”Limited capabilities (FMT_LIM.1[Loader])” the following policy is enforced: *The TSF prevents deploying the Loader functionality after switching to LifeCycleState.Release.*

6.1.1.2 SFRs for Augmentation Package “Loader Package2”

6.1.1.2.1 Subjects

Subject	DownloadUsr	Download User
Info	User Role to download data, verify data and erase data in memory areas.	

Subject	KeyUsr	Key Change User
Info	User Role to update and verify keys.	

Subject	FirewallUsr	Firewall User
Info	User Role to change firewall settings of memory areas.	

Subject	DeveloperModeUsr	Developer Mode User
Info	User Role to switch the LifeCycle to PreRelease.	

Subject	ProductionModeUsr	Production Mode User
Info	User Role to switch the LifeCycle to Release.	

Subject	FLASHUsr	FLASH User
Info	User Role to set the logical available size of FLASH memory.	

Subject	CardOS	Card Operating System
Info	The Card Operationg System.	

6.1.1.2.2 Objects

Object	LifeCycleState	Life Cycle State of the Loader
Info	Life Cycle of the Loader.	
Operation	switch	Switch from CardAppMgmt to Pre-Release, from Pre-Release to CardAppMgmt or from CardAppMgmt to Release.
Attribute	CardAppMgmt	Initial LifeCycle of the TOE, Card and Application Management which allows download operations.
Attribute	Pre-Release	LifeCycle Pre-Release in which the previously downloaded code can be executed. Furthermore it is possible in to switch the Life-Cycle back to LifeCycle CardAppMgmt.
Attribute	Release	LifeCycle Release in which no download operations can be per-formed.

Object	Keys	Keys
Info	Cryptographic keys used to identify users.	
Operation	update	Update and verify a key.
Attribute	Permissions	The permissions assosicated with one key to identify subjects.

Object	FirewallSettings	Firewall Settings
Info	Firewall border settings for memory segments.	
Operation	change	Change the Firewall Settings.

Object	MemorySegment	Memory Segment in FLASH
Info	A memory segment to which data or code can be downloaded.	
Operation	download	Download, verify or erase data within a memory segment.

Object	FLASHSize	FLASH size
Info	The logical available size of FLASH memory.	
Operation	set	Set the available size of FLASH memory.

FTP_ITC.1	**Inter-TSF trusted channel**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FTP_ITC.1.1	The TSF shall provide a communication channel between itself and *DownloadUsr, KeyUsr, FirewallUsr, DeveloperModeUsr, ProductionModeUsr and FLASHUsr* that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2	The TSF shall permit *another trusted IT product* to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for *deploying Loader functionality as described in FDP_ACF.1[Loader]*.
FDP_UCT.1	**Basic data exchange confidentiality**
Hierarchical-To	No other components.
Dependencies	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_UCT.1.1	The TSF shall enforce the *Loader SFP* to *receive* user data in a manner protected from unauthorised disclosure.
FDP_UIT.1	**Data exchange integrity**
Hierarchical-To	No other components.
Dependencies	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]
FDP_UIT.1.1	The TSF shall enforce the *Loader SFP* to *receive* user data in a manner protected from *modification, deletion, insertion* errors.

FDP_UIT.1.2	The TSF shall be able to determine on receipt of user data, whether *modification, deletion, insertion* has occurred.
FDP_ACC.1[Loader]	**Subset access control - Loader**
Hierarchical-To	No other components.
Dependencies	FDP_ACF.1 Security attribute based access control.
FDP_ACC.1.1[Loader]	The TSF shall enforce the *Loader SFP* on 1. *the subjects DownloadUsr, KeyUsr, FirewallUsr, DeveloperModeUsr, ProductionModeUsr, FLASHUsr, and CardOS*, 2. *the objects user data in LifeCycleState, Keys, FirewallSettings, MemorySegment and FLASHSize*, 3. *the operation deployment of Loader.*
FDP_ACF.1[Loader]	**Security attribute based access control - Loader**
Hierarchical-To	No other components.
Dependencies	FMT_MSA.3 Static attribute initialisation.
FDP_ACF.1.1[Loader]	FDP_ACF.1.1 The TSF shall enforce the *Loader SFP* to objects based on the following: 1. *the subjects DownloadUsr, KeyUsr, FirewallUsr, DeveloperModeUsr, ProductionModeUsr, FLASHUsr, and CardOS* with security attributes *none*, 2. *the objects user data in Flash memory with security attributes LifeCycleState, Keys, FirewallSettings, MemorySegment and FLASHSize.*
FDP_ACF.1.2[Loader]	FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: ACF12:DMU:LCS1 *The DeveloperModeUsr is allowed to perform LifeCycleState.switch from LifeCycleState.CardAppMgmt to LifeCycleState.Pre-Release.* ACF12:PMU:LCS1 *The ProductionModeUsr is allowed to perform LifeCycleState.switch from LifeCycleState.CardAppMgmt to LifeCycleState.Release.* ACF12:COS:LCS1 *The CardOS is allowed to perform LifeCycleState.switch from LifeCycleState.Pre-Release to LifeCycleState.CardAppMgmt.*
FDP_ACF.1.3[Loader]	FDP_ACF.1.3 The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: ACF13:DLU:MS1 *The DownloadUsr is allowed to perform MemorySegment.download if LifeCycleState.CardAppMgmt grants this right.* ACF13:KU:K1 *The KeyUsr is allowed to perform Keys.update if LifeCycleState.CardAppMgmt grants this right.* ACF13:FWU:FS1 *The FirewallUsr is allowed to perform FirewallSettings.change if LifeCycleState.CardAppMgmt grants this right.*

ACF13:FU:F1 *The FLASHUsr is allowed to perform FLASHSize.set if LifeCycleState.CardAppMgmt grants this right.*

FDP_ACF.1.4[Loader] The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *as stated in SFR FMT_LIM.2[Loader]*.

6.1.1.3 SFRs for Augmentation Package "TDES"

The TOE shall meet the requirement "Cryptographic operation (FCS_COP.1)" and "Cryptographic key destruction (FCS_CKM.4)" as specified below.

FCS_COP.1[TDES_HW] Cryptographic operation - TDES - Hardware Support

Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1[TDES_HW]	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *TDES* in *ECB mode* and cryptographic key sizes *168 bit* that meet the following *NIST SP 800-67 [12], NIST SP 800-38A [10]*.

FCS_COP.1[TDES_SW] Cryptographic operation - TDES - Software Support

Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1[TDES_SW]	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *TDES in ECB mode, CBC mode, CBC-MAC, Retail-MAC mode and CMAC mode* and cryptographic key sizes *168 bit* that meet the following *NIST SP 800-67 (TDES) [12], NIST SP 800-38A (ECB and CBC mode) [10], ISO 9797-1, Algorithm 1 (CBC-MAC mode) [31], ISO 9797-1, Algorithm 3 (Retail-MAC) [31] and NIST SP 800-38B (CMAC mode) [11]*.

FCS_CKM.4[TDES_SW] Cryptographic key destruction - TDES - Software

Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4.1[TDES_SW]	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *flushing of key registers* that meets the following: *none*.
Application Note:	The *N7021 VA* provides the smartcard embedded software with library calls to perform various cryptographic algorithms that involve keys (e.g AES, DES, etc.). Through the parameters

of the library calls the smartcard embedded software provides keys for the cryptographic algorithms. To perform its cryptographic algorithms the library copies these keys, or a transformation thereof, to the working-buffer (supplied by the smartcard embedded software) and/or the memory/special function registers of the *N7021 VA*. Depending upon the algorithm the library either overwrites these keys before returning control to the smartcard embedded software or provides a library call to through which the smartcard embedded software can clear these keys. In the case of a separate library call to clear keys the guidance instructs the smartcard embedded software when/how this call should be used.

Note: The *N7021 VA* provides the embedded software with functionality for key destruction for FCS_COP.1[TDES_SW]. Clearing of keys that are provided by the smartcard embedded software to the *N7021 VA* is the responsibility of the smartcard embedded software.

6.1.1.4 SFRs for Augmentation Package "AES"

The TOE shall meet the requirement "Cryptographic operation (FCS_COP.1)" and "Cryptographic key destruction (FCS_CKM.4)" as specified below.

FCS_COP.1[AES_HW]	**Cryptographic operation - AES - Hardware Support**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1[AES_HW]	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *AES in ECB mode* and cryptographic key sizes *128 bit, 192 bit, 256 bit* that meet the following: *FIPS 197 [8], NIST SP 800-38A [10].*

FCS_COP.1[AES_SW]	**Cryptographic operation - AES - Software Support**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1[AES_SW]	The TSF shall perform *encryption and decryption* in accordance with a specified cryptographic algorithm *AES in ECB mode, CBC mode, CBC-MAC mode and CMAC mode* and cryptographic key sizes *128 bit, 192 bit and 256 bit* that meet the following: *FIPS 197 [8], NIST SP 800-38A (ECB and CBC mode) [10], ISO 9797-1, Algorithm 1 (CBC-MAC mode) [31], and NIST SP 800-38B (CMAC mode) [11].*

FCS_CKM.4[AES_SW]	**Cryptographic key destruction - AES - Software**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1[AES_SW]	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method *flushing of key registers* that meets the following: *none*.
Note:	The *N7021 VA* provides the embedded software with functionality for key destruction for FCS_COP.1[AES_SW]. Clearing of keys that are provided by the smartcard embedded software to the *N7021 VA* is the responsibility of the smartcard embedded software.

In compliance with Application Note 12 in the PP [26] the following section defines additional SFRs related to cryptographic functionality and access control functionality, which are required by this Security Target, but not by the PP [26].
As required by Application Note 14 in the PP [26] the secure state is described in Section **??** in the rationale for SF.OPC.
Regarding Application Note 15 in the PP [26] generation of additional audit data is not defined for requirements FRU_FLT.2 and FPT_FLS.1.
As required by Application Note 19 in the PP [26] the automatic response of the TOE is described in Section **??** in the rationale for SF.PHY.

6.1.2 Additional SFRs regarding Cryptographic Support

The TOE shall meet the requirement ”TSF Testing (FPT_TST.1)” as specified below.

FCS_COP.1[AES_PUF]	**Cryptographic operation - PUF based AES**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction.
FCS_COP.1.1[AES_PUF]	The TSF shall perform *decryption and encryption* in accordance with a specified cryptographic algorithm *AES in CBC mode* and cryptographic key size *128 bits* that meets the following: *FIPS 197 [8], NIST SP 800-38A [10]*.

FCS_COP.1[MAC_PUF]	**Cryptographic operation - PUF based MAC**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction.
FCS_COP.1.1[MAC_PUF]	The TSF shall perform *calculation of CBC-MAC values used for PUF authentication* in accordance with a specified cryptographic algorithm *AES in CBC-MAC mode* and cryptographic key size *128 bits* that meets the following: *FIPS 197 [8], NIST SP 800-38A [10] and ISO/IEC 9797-1 (MAC algorithm 1) [31]*.

FCS_CKM.1[PUF]	**Cryptographic Key Generation - PUF**
Hierarchical-To	No other components.
Dependencies	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
FCS_CKM.1.1[PUF]	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *key derivation function based on PUF* and specified cryptographic key sizes *128 bits* that meet the following: *[32]*.

FCS_CKM.4[PUF]	**Cryptographic Key Destruction - PUF**
Hierarchical-To	No other components.
Dependencies	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4.1[PUF]	The TSF shall destroy cryptographic keys *derived by key derivation function based on PUF* in accordance with a specified cryptographic key destruction method *flushing of key registers* that meets the following: *none*.

6.1.3 Additional SFRs regarding Protection of TSF

The TOE shall meet the requirement ”TSF Testing (FPT_TST.1)” as specified below.

FPT_TST.1	**TSF Testing**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FPT_TST.1.1	The TSF shall run a suite of self tests *at the request of the authorised user* to demonstrate the correct operation of • *the active shielding* • *the sensors*
FPT_TST.1.2	The TSF shall provide authorised users with the capability to verify the integrity of *Special Function Registers holding static values loaded during start-up*.
FPT_TST.1.3	The TSF shall provide authorised users with the capability to verify the integrity of *stored TSF executable code*.

6.1.4 Additional SFRs regarding Security Management

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

FMT_SMF.1[SW]	**Specification of Management Functions (Software)**
Hierarchical-To	No other components.

Dependencies	No dependencies.
FMT_SMF.1.1[SW]	The TSF shall be capable of performing the following management functions: • *Performing a System Reset* • *Performing a Security Reset* • *Terminating the IC* • *Getting the state of the Error Counter* • *Getting the state of the Delay Latch* • *Reading out the FabKey area*

6.1.5 Additional SFRs regarding User Data Protection

The TOE shall meet the requirement ”Subset Residual Information Protection (FDP_RIP.1)” as specified below.

FDP_RIP.1[SW]	**Subset Residual Information Protection**
Hierarchical-To	No other components.
Dependencies	No dependencies.
FDP_RIP.1.1[SW]	The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource from* the following objects: *all cryptographic assets (such as keys, ciphers, plain text) stored temporarily in memory used by the TOE.*
Note:	The TSF ensures that, upon exit from each function, with the exception of input parameters, return values or locations where it is explicitly documented that values remain at specific addresses, any memory resources used by that function that contained temporary or secret values are cleared.

6.1.6 Additional SFRs regarding Access Control

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

The Security Function Policy (SFP) **Hardware Access Control Policy** uses the definitions defined in the following sections. Thereby, subjects, objects and attributes are defined in a semi-formal tabular way. Each of them is equipped with a unique label shown in the second column of each table’s header. Subjects and object are provided with a title and a descriptive block in addition. Operations can belong to objects (in that case contained in the first column) or to attributes (in that case contained in the second column).

6.1.6.1 Subjects

Subject	SSM_Code	Code run in Super System Mode
Info	Parts of the Boot Software and the Firmware Interface as part of the IC Dedicated Support Software, executed as instructions by the CPU.	

Subject	SSM-CM_Code	Code run in Configuration Mode
Info	Parts of the Boot Software and the Firmware Interface as part of the IC Dedicated Support Software, executed as instructions by the CPU.	

Subject	SSM-TM_Code	Code run in Test Mode
Info	The Test Software as the IC Dedicated Test Software, executed as instructions by the CPU.	

Subject	SM-A_Code	Code run in System Mode Card A
Info	Parts of the IC Dedicated Support Software and parts of the Security IC Embedded Software (System Mode Customer Code) in Card A, executed as instructions by the CPU.	

Subject	SM-B_Code	Code run in System Mode Card B
Info	Parts of the IC Dedicated Support Software and parts of the Security IC Embedded Software (System Mode Customer Code) in Card B, executed as instructions by the CPU.	

Subject	UM-A_Code	Code run in User Mode Card A
Info	The Security IC Embedded Software (User Mode Customer Code) in Card A, executed as instructions by the CPU.	

Subject	UM-B_Code	Code run in User Mode Card B
Info	The Security IC Embedded Software (User Mode Customer Code) in Card B, executed as instructions by the CPU.	

Subject	SM-A_NXP-Code	NXP Code run in System Mode Card A
Info	Parts of the IC Dedicated Support Software and parts of the System Mode Card A Operating System provided by NXP, executed as instructions by the CPU.	

Subject	UM-A_SecureBox-Code	Code run in the NXP Secure User Mode Box in Card A
Info	The Software executed in User Mode within the NXP Secure User Mode Box in Card A, executed as instructions by the CPU.	

6.1.6.2 Objects/Operations/Security Attributes related to Data in Memories

Object	All_Mem	Memory
Info	All data and code segments.	
Operation	access	Access memory segments.
Attribute	enable	Enable r/w access via SFR_MIRROR.

Object	SSM_Data_Seg	Super System Mode Data Segment
Info	Data segment that contains data of the Super System Mode Code.	
Operation	access	Access data.

Object	SM-A_Data_Seg	System Mode Card A Data Segment
Info	Data segment that contains data of the System Mode Card A Code.	
Operation	access	Access data.
Attribute	shared	Enable sharing of parts of the segment via SFR_DYN_SEG, SFR_FRAM, SFR_CRAM, or SFR_SM_MemCfg.

Object	SM-B_Data_Seg	System Mode Card B Data Segment
Info	Data segment that contains data of the System Mode Card B Code.	
Operation	access	Access data.
Attribute	shared	Enable sharing of parts of the segment via SFR_DYN_SEG, SFR_FRAM, SFR_CRAM, or SFR_SM_MemCfg.

Object	UM-A_Data_Seg	User Mode Card A Data Segment
Info	Data segment that contains data of the User Mode Card A Code.	
Operation	access	Access data.
Attribute	shared	Enable sharing of parts of the segment via SFR_DYN_SEG, SFR_FRAM, SFR_CRAM, or SFR_SM_MemCfg.

Object	UM-B_Data_Seg	User Mode Card B Data Segment
Info	Data segment that contains data of the User Mode Card B Code.	
Operation	access	Access data.

Object	UM-B_Data_Seg	User Mode Card B Data Segment
Attribute	shared	Enable sharing of parts of the segment via SFR_DYN_SEG, SFR_FRAM, SFR_CRAM, or SFR_SM_MemCfg.

6.1.6.3 Objects/Operations/Security Attributes related to Code in Memories

Object	SSM_Code_Seg	Super System Mode Code Segment
Info	Contains the code of the TOE that runs with Super System Mode priviledge.	
Operation	execute	Execute code.

Object	SM-A_Code_Seg	System Mode Card A Code Segment
Info	Contains the code of the Card A that runs with System Mode priviledge.	
Operation	execute	Execute code.
Attribute	shared	Enable sharing of parts of the segment via SFR_DYN_SEG.

Object	SM-B_Code_Seg	System Mode Card B Code Segment
Info	Contains the code of the Card B that runs with System Mode priviledge.	
Operation	execute	Execute code.
Attribute	shared	Enable sharing of parts of the segment via SFR_DYN_SEG.

Object	UM-A_Code_Seg	User Mode Card A Code Segment
Info	Contains the code of the Card A that runs with User Mode priviledge.	
Operation	execute	Execute code.
Attribute	shared	Enable sharing of parts of the segment via SFR_DYN_SEG.

Object	UM-B_Code_Seg	User Mode Card B Code Segment
Info	Contains the code of the Card B that runs with User Mode priviledge.	
Operation	execute	Execute code.
Attribute	shared	Enable sharing of parts of the segment via SFR_DYN_SEG.

6.1.6.4 Objects/Operations/Security Attributes related to Special Function Registers

Object	SFR_CardCfg	Special Function Registers related to Configuration of the Card
Info	Group of Special Function Registers related to the configuration of the card. For example Special Function Registers to set the size of Card A and Card B.	
Operation	read	Read base address or size.

Object	SFR_CardCfg	Special Function Registers related to Configuration of the Card
Operation	write	Write base address or size.

Object	SFR_SSM_MemCfg	Special Function Registers related to Super System Mode Memory Segment Configuration
Info	Group of Special Function Registers to configure the data and code segments for Super System Mode.	
Operation	read	Read base address, size, or control.
Operation	write	Write a base address, size, or control.

Object	SFR_SM_MemCfg	Special Function Registers related to System Mode Memory Segment Configuration
Info	Group of Special Function Registers to configure the data and code segments for System Mode.	
Operation	read	Read base address, size, or control.
Operation	write	Write a base address, size, or control.

Object	SFR_UM_MemCfg	Special Function Registers related to User Mode Memory Segment Configuration
Info	Group of Special Function Registers to configure the data and code segments for User Mode.	
Operation	read	Read base address, size, or control.
Operation	write	Write a base address, size, or control.

Object	SFR_PAC	Special Function Registers related to Peripheral Access Control
Info	Group of Special Function Registers defining the owner of a peripheral.	
Operation	read	Read a configuration setting / value.
Operation	write	Write a configuration setting / value.

Object	SFR_DYN_SEG	Special Function Registers related to Dynamic Segments
Info	Group of Special Function Registers used to set up dynamic segments.	
Operation	read	Read address and configuration setting / value.
Operation	write	Write address and configuration setting / value.
Attribute	ownership	Define the owner of the dynamic segment via SFR_DYN_SEG.

Object	SFR_FRAM	Special Function Registers related to the FRAM
Info	Group of Special Function Registers used to define the FRAM segment.	
Operation	read	Read address and configuration setting / value.
Operation	write	Write address and configuration setting / value.
Attribute	ownership	Request of ownership for the Public Key Crypto Coprocessor via SFR_PAC.

Object	SFR_CRAM	Special Function Registers related to the CRAM
Info	Group of Special Function Registers used to define the CRAM segment.	
Operation	read	Read address and configuration setting / value.
Operation	write	Write address and configuration setting / value.
Attribute	ownership	Request of ownership for the Communication Interface via SFR_PAC.

Object	SFR_MIRROR	Special Function Registers enabling the Mirror Segments
Info	Group of Special Function Registers enabling the mirror segments.	
Operation	read	Read a configuration setting.
Operation	write	Write a configuration setting.

Object	SFR_Testing	Special Function Registers related to Testing
Info	Group of Special Function Registers reserved for testing purposes.	
Operation	read	Read a configuration setting / value.
Operation	write	Write a configuration setting / value.

Object	SFR_HWComp	Special Function Registers related to Hardware Components
Info	Group of Special Function Registers used to utilize the following hardware components: • AES Coprocessor • DES Coprocessor • Public Key Crypto Coprocessor • CRC Coprocessor • Physical Random Number Generator • Communication Interface • Timer	

Object	SFR_HWComp	Special Function Registers related to Hardware Components
Operation	read	Read a configuration setting / value.
Operation	write	Write a configuration setting / value.
Attribute	ownership	Request of ownership for one of the hardware components via SFR_PAC.

6.1.6.5 Access Rules

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

FDP_ACC.1[MEM]	**Subset Access Control (Memories)**
Hierarchical-To	No other components.
Dependencies	FDP_ACF.1 Security attribute based access control.
FDP_ACC.1.1[MEM]	The TSF shall enforce the *Hardware Access Control Policy* on *all code running on the TOE, all memories and all memory operations*.
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. A denied read or write access or read/write to a non-existing memory address is treated as a security violation and will trigger a Security Reset.

FDP_ACC.1[SFR]	**Subset Access Control (Special Function Registers)**
Hierarchical-To	No other components.
Dependencies	FDP_ACF.1 Security attribute based access control.
FDP_ACC.1.1[SFR]	The TSF shall enforce the *Hardware Access Control Policy* on *all code running on the TOE, all Special Function Registers and all Special Function Register operations.*
Application Note:	The Hardware Access Control Policy shall be enforced by implementing hardware access control to each Special Function Register. For every access the TOE mode is used to determine if the access shall be granted or denied. A denied read or write access or read/write to a non-existing Special Function Registers is treated as a security violation and will trigger a Security Reset.

FDP_ACC.1[SUB]	**Subset Access Control (Secure User Mode Box)**
Hierarchical-To	No other components.
Dependencies	FDP_ACF.1 Security attribute based access control.
FDP_ACC.1.1[SUB]	The TSF shall enforce the *NXP Secure User Mode Box Policy* on *all code running on the TOE within the Secure User Mode Box, all memories and all memory operations*.

Application Note: The Access Control Policy shall be enforced by implementing a Memory Management Unit, which maps virtual addresses to physical addresses together with an NXP System Mode OS in logical card A configuring the borders of the NXP Secure User Mode Box in logical card A. 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. A denied read or write access or read/write to a non-existing memory address is treated as a security violation and will trigger a Security Reset.

The following access control rules are defined in a semi-formal way, i.e. each rule is provided with a unique label and each rule exactly identifies the subject (via its label defined in section 6.1.6.1), object (via its label defined in the sections 6.1.6.2, 6.1.6.3 and 6.1.6.4, respectively) and operation (added to the associated operation via "."). For operations with explicit authorized access, the related attribute is referenced (as shown via a hyperlink to the unique label of the attribute associated to the operation via ".").

FDP_ACF.1[MEM]	**Security Attribute Based Access Control (Memories)**
Hierarchical-To	No other components.
Dependencies	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization
FDP_ACF.1.1[MEM]	The TSF shall enforce the *Hardware Access Control Policy* to objects based on the following: *all subjects and objects and the attributes themselves defined as the objects SFR_MIRROR, SFR_DYN_SEG, SFR_FRAM, SFR_CRAM, and SFR_SM_MemCfg.*
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:
ACF12.MEM:data	*Code running in a certain mode (SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, or UM-B_Code respectively) is allowed to access the data segment of this mode (is allowed to perform SSM_Data_Seg.access, SM-A_Data_Seg.access, SM-B_Data_Seg.access, UM-A_Data_Seg.access, or UM-B_Data_Seg.access respectively).*
ACF12.MEM:code	*Code running in a certain mode (SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, or UM-B_Code respectively) is allowed to execute code from the code segment of this mode (is allowed to perform SSM_Code_Seg.execute, SM-A_Code_Seg.execute, SM-B_Code_Seg.execute, UM-A_Code_Seg.execute, or UM-B_Code_Seg.execute respectively).*
FDP_ACF.1.3[MEM]	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules:
ACF13.MEM:mirror	*The SSM_Code, SSM-CM_Code, and SSM-TM_Code is allowed to perform All_Mem.access if enabled via SFR_MIRROR.*
ACF13.MEM:data	*Code running in a certain mode (SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, or UM-B_Code) can access data of another modes*

segment (is allowed to perform SM-A_Data_Seg.access, SM-B_Data_Seg.access, UM-A_Data_Seg.access, or UM-B_Data_Seg.access) if this other mode shares the data with the currently running mode via SFR_DYN_SEG, SFR_FRAM, SFR_CRAM, or SFR_SM_MemCfg.

ACF13.MEM:code *Code running in a certain mode (SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, or UM-B_Code) can execute code of another modes segment (is allowed to perform SM-A_Code_Seg.execute, SM-B_Code_Seg.execute, UM-A_Code_Seg.execute, or UM-B_Code_Seg.execute) if this other mode shares the code with the currently running mode via SFR_DYN_SEG.*

FDP_ACF.1.4[MEM] The TSF shall explicitly deny access of subjects to objects based on the rules: *none*.

FDP_ACF.1[SFR] Security Attribute Based Access Control (Special Function Registers)

Hierarchical-To No other components.

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

FDP_ACF.1.1[SFR] The TSF shall enforce the *Hardware Access Control Policy* to objects based on the following: *all subjects and objects and the attributes itself defined as the objects SFR_DYN_SEG and SFR_PAC.*

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:

ACF12.SFR:CardCfg *The SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, and SM-B_Code are allowed to perform SFR_CardCfg.read and SSM-CM_Code, and SSM-TM_Code are allowed to perform SFR_CardCfg.write.*

ACF12.SFR:SSM_MemCfg *The SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, and SM-B_Code are allowed to perform SFR_SSM_MemCfg.read and SSM-CM_Code, and SSM-TM_Code are allowed to perform SFR_SSM_MemCfg.write.*

ACF12.SFR:SM_MemCfg *The SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, and SM-B_Code are allowed to perform SFR_SM_MemCfg.read and SSM-CM_Code, SSM-TM_Code, SM-A_Code, and SM-B_Code are allowed to performSFR_SM_MemCfg.write.*

ACF12.SFR:UM_MemCfg *The SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, and UM-B_Code are allowed to perform SFR_UM_MemCfg.read and SSM-CM_Code, SSM-TM_Code, SM-A_Code, and SM-B_Code, are allowed to perform SFR_UM_MemCfg.write.*

ACF12.SFR:PAC *The SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, and UM-B_Code are allowed to perform SFR_PAC.read and SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, and SM-B_Code are allowed to perform SFR_PAC.write.*

ACF12.SFR:DYN_SEG *The SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, and UM-B_Code are allowed to perform SFR_DYN_SEG.read and SSM_Code, SSM-CM_Code, and SSM-TM_Code are allowed to perform SFR_DYN_SEG.write.*

ACF12.SFR:MIRROR *The SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, and UM-B_Code are allowed to perform SFR_MIRROR.read and SSM_Code, SSM-CM_Code, and SSM-TM_Code are allowed to perform SFR_MIRROR.write.*

ACF12.SFR:Testing *The SSM-CM_Code, and SSM-TM_Code, is allowed to perform SFR_Testing.read and SFR_Testing.write.*

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

ACF13.SFR:DYN_SEG *The SM-A_Code and SM-B_Code are allowed to perform SFR_DYN_SEG.write if SFR_DYN_SEG.ownership grants this right.*

ACF13.SFR:FRAM *The SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, and UM-B_Code are allowed to perform SFR_FRAM.read and SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, and SM-B_Code are allowed to perform SFR_FRAM.write, if SFR_FRAM.ownership grants this right.*

ACF13.SFR:CRAM *The SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, SM-B_Code, UM-A_Code, and UM-B_Code are allowed to perform SFR_CRAM.read and SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, and SM-B_Code are allowed to perform SFR_CRAM.write, if SFR_CRAM.ownership grants this right.*

ACF13.SFR:HWComp *Read or write access to Special Function Registers of a hardware component is only possible if SFR_HWComp.ownership grants this right.*

FDP_ACF.1.4[SFR] The TSF shall explicitly deny access of subjects to objects based on the rules: *none*.

FDP_ACF.1[SUB] Security Attribute Based Access Control (Secure User Mode Box)

Hierarchical-To No other components.

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

FDP_ACF.1.1[SUB] The TSF shall enforce the *NXP Secure User Mode Box Policy* to objects based on the following: *all subjects and objects and the attributes themselves defined as the objects SFR_DYN_SEG, SFR_FRAM and SFR_CRAM.*

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

ACF12.SUB *In addition to the restrictions provided by the "Hardware Access Control Policy", the "NXP Secure User Mode Box Policy" implements further NXP-defined restrictive default values for accessing SFR_DYN_SEG, SFR_FRAM and SFR_CRAM.*

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

ACF13.SUB *The UM-A_SecureBox-Code is allowed to change a restricted set of attributes of SFR_DYN_SEG, SFR_FRAM and SFR_CRAM.*

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

ACF13.SUB *The UM-A_SecureBox-Code is not allowed to change its access rights to SFR_DYN_SEG, SFR_FRAM and SFR_CRAM.*

6.1.6.6 Implications of the Hardware Access Control Policy

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

- Code executed in Super System Mode is quite powerful and used to configure and test the TOE.
- Code executed in the System Mode can administrate the configuration of Memory Management Unit, because it has access to the respective Special Function Registers.
- Code executed in the User Mode hardly administrate the configuration of the TOE, because it has very limited access to the related Special Function Registers.

FMT_MSA.3[MEM]	**Static Attribute Initialization (Memories)**
Hierarchical-To	No other components.
Dependencies	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1[MEM]	The TSF shall enforce the *Hardware Access Control Policy* to provide *restrictive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2[MEM]	The TSF shall allow *no subject* to specify alternative initial values to override the default values when an object or information is created.
Application Note:	Restrictive means that the reset values of the Special Function Registers which are security attributes are set to zero. 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.

FMT_MSA.3[SFR]	**Static Attribute Initialization (Special Function Registers)**
Hierarchical-To	No other components.
Dependencies	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1[SFR]	The TSF shall enforce the *Hardware Access Control Policy* to provide *restrictive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2[SFR]	The TSF shall allow *no subject* to specify alternative initial values to override the default values when an object or information is created.
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.

FMT_MSA.3[SUB]	**Static Attribute Initialization (Secure User Mode Box)**
Hierarchical-To	No other components.

Dependencies	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1[SUB]	The TSF shall enforce the *NXP Secure User Mode Box Policy* to provide *restrictive* default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2[SUB]	The TSF shall allow *no subject* to specify alternative initial values to override the default values when an object or information is created.
Application Note:	Restrictive means that the reset values of the Special Function Registers which are security attributes are set to zero. 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.

FMT_MSA.1[MEM]	**Management of Security Attributes (Memories)**
Hierarchical-To	No other components.
Dependencies	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1[MEM]	The TSF shall enforce the *Hardware Access Control Policy* to restrict the ability to *modify* the security attributes *defined as the objects SFR_MIRROR, SFR_DYN_SEG, SFR_FRAM, SFR_CRAM, and SFR_SM_MemCfg* to *SSM_Code, SSM-CM_Code, SSM-TM_Code, SM-A_Code, and SM-B_Code*.

FMT_MSA.1[SFR]	**Management of Security Attributes (Special Function Registers)**
Hierarchical-To	No other components.
Dependencies	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1[SFR]	The TSF shall enforce the *Hardware Access Control Policy* to restrict the ability to *modify* the security attributes *defined in the Special Function Registers* to *code executed in a TOE mode which has write access to the respective Special Function Registers.*

FMT_MSA.1[SUB]	**Management of Security Attributes (Secure User Mode Box)**
Hierarchical-To	No other components.
Dependencies	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] FMT_SMR.1 Security roles FMT_SMF.1 Specification of Management Functions
FMT_MSA.1.1[SUB]	The TSF shall enforce the *NXP Secure User Mode Box Policy* to restrict the ability to *modify* the security attributes *defined as the objects SFR_DYN_SEG, SFR_FRAM, and SFR_CRAM* to *SSM_Code, SSM-CM_Code, SSM-TM_Code, and SM-A_NXP-Code*.

FMT_SMF.1[HW]	**Specification of Management Functions (Hardware)**
Hierarchical-To	No other components.

Dependencies	No dependencies.
FMT_SMF.1.1[HW]	The TSF shall be capable of performing the following management functions:
SMF11.HW:USR	*Change of TOE mode to lower privileged mode by calling one of the following instructions: USR or EUSR and SYSACK.*
SMF11.HW:SYS	*Change of TOE mode to higher privileged mode by calling one of the following instructions: SYS or ESYS and SYSACK.*
SMF11.HW:INT	*Change of TOE mode by invoking an interrupt.*
SMF11.HW:RETI	*Change of TOE mode by finishing an interrupt (with instruction RETI).*
SMF11.HW:TPDC	*Temporary disabling and enabling of security functionality using PDC (see Table 1.3).*
SMF11.HW:PPDC	*Permanently disabling and enabling of security functionality using PDC (see Table 1.3).*

Application Note: The iteration of FMT_MSA.1 with the dependency to FMT_SMF.1 may imply a separation of the Specification of Management Functions. However, iteration of FMT_SMF.1 is not needed for hardware access control (FMT_MSA.1[MEM] and FMT_MSA.1[SFR]) because all management functions rely on the same features implemented in the hardware.

6.2 Security Assurance Requirements

Table 6.46 lists all security assurance requirements that are valid for this Security Target. These security assurance requirements are defined in the PP ”Security IC Platform Protection Profile” [26] and/or in CC part [5] for EAL6, except for requirements ASE_TSS.2 and ALC_FLR.1, which are augmentations of this Security Target to EAL6, see section 2.2.

ASE_TSS.2 is an augmentation in this Security Target to give architectural information on the security functionality of the TOE. ALC_FLR.1 is 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.

In compliance with Application Note 22 in the PP [26] the third column in Table 6.46 shows, which security assurance requirements are added to this Security Target compared to the PP [26]. In this context, entry ”EAL6 / PP” means, that the requirement is defined in both, CC part [5] for EAL6 and the PP [26], entry ”EAL6” means, that the requirement is defined in CC part [5] for EAL6 but not in the PP [26], and entry ”ST” means, that the requirement is defined neither in CC part [5] for EAL6 nor in the PP [26], but in this Security Target.

All refinements of the security assurance requirements in the PP [26], which must be adapted for EAL6, are described in section 6.2.1.

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 of the implementation representation of the TSF	EAL6
ADV_INT.3	Minimally complex internals	EAL6

SAR	Title	Required by
ADV_TDS.5	Complete semiformal modular design	EAL6
ADV_SPM.1	Formal TOE security policy model	EAL6
AGD_OPE.1	Operational user guidance	EAL6 / PP
AGD_PRE.1	Preparative procedures	EAL6 / PP
ALC_CMC.5	Advanced support	EAL6
ALC_CMS.5	Development tools CM coverage	EAL6
ALC_DEL.1	Delivery procedures	EAL6 / PP
ALC_DVS.2	Sufficiency of security measures	EAL6 / PP
ALC_FLR.1	Basic flaw remediation	ST
ALC_LCD.1	Developer defined life-cycle model	EAL6 / PP
ALC_TAT.3	Compliance with implementation standards – all parts	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	Rigorous analysis of coverage	EAL6
ATE_DPT.3	Testing: modular design	EAL6
ATE_FUN.2	Ordered Functional testing	EAL6
ATE_IND.2	Independent testing – sample	EAL6 / PP
AVA_VAN.5	Advanced methodical vulnerability analysis	EAL6 / PP

Tab. 6.46: SARs for this ST

In the set of assurance components chosen for EAL6, the assignment appears only in ADV_SPM.1. The assignment for ADV_SPM.1 is defined below.

ADV_SPM.1 Formal TOE security policy model

Dependencies: ADV_FSP.4

Developer action elements:

ADV_SPM.1.1D The developer shall provide a formal security policy model for the

- *Limited Capability and Availability Policy (FMT_LIM.1[HW] and FMT_LIM.2[HW]),*
- *Hardware Access Control Policy (FDP_ACC.1[MEM], FDP_ACC.1[SFR], FDP_ACF.1[MEM], FDP_ACF.1[SFR], FMT_MSA.1[MEM], FMT_MSA.1[SFR], FMT_MSA.3[MEM], FMT_MSA.3[SFR] and FMT_SMF.1[HW]),*
- *NXP Secure User Mode box policy (FDP_ACC.1[SUB], FDP_ACF.1[SUB], FMT_MSA.1[SUB] and FMT_MSA.3[SUB]),*

- *Loader SFP (FDP_ACC.1[Loader], FDP_ACF.1[Loader], FDP_UCT.1, FDP_UIT.1, FMT_LIM.1[Loader], FMT_LIM.2[Loader] and FTP_ITC.1).*

Additionally we model security policy related parts of the following Security Functional Requirements: FAU_SAS.1[HW] and FPT_FLS.1, FMT_SMF.1[SW] and FPT_TST.1.

ADV_SPM.1.2D For each policy covered by the formal security policy model, the model shall identify the relevant portions of the statement of SFRs that make up that policy.

ADV_SPM.1.3D The developer shall provide a formal proof of correspondence between the model and any formal functional specification.

ADV_SPM.1.4D The developer shall provide a demonstration of correspondence between the model and the functional specification.

6.2.1 Refinements of the TOE Security Assurance Requirements

In compliance to Application Note 23 in the PP [26] this Security Target has to conform to all refinements of the security assurance requirements in the PP [26]. These refinements are defined for the security assurance requirements of EAL4. Thus, some of these refinements must be adapted to security assurance requirements of higher levels according to EAL6 as claimed in this Security Target. All other security assurance requirements defined in this Security Target and in particular the augmentations to EAL6 supplement and extent the security assurance requirements in the PP [26] and can be added without contradictions.
Table 6.47 lists all security assurance requirements that are refined in the PP [26] based on their definitions is CC part [5] and their effect on this Security Target.

Refined SAR in PP [26]	Effect on Security Target
ADV_ARC.1	SAR same as in PP, refinement valid without change
ADV_FSP.4	SAR moves to ADV_FSP.5, refinement valid without change
ADV_IMP.1	ADV_IMP.2, refinement valid without change
AGD_OPE.1	SAR same as in PP, refinement valid without change
AGD_PRE.1	SAR same as in PP, refinement valid without change
ALC_CMC.4	SAR moves to ALC_CMC.5, refinement valid without change
ALC_CMS.4	SAR moves to ALC_CMS.5, refinement valid without change
ALC_DEL.1	Same as in PP, refinement valid without change
ALC_DVS.2	Same as in PP, refinement valid without change
ATE_COV.2	SAR moves to ATE_COV.3, refinement valid without change
AVA_VAN.5	Same as in PP, refinement valid without change

Tab. 6.47: SARs refined in the PP [26] and their effect on this ST

6.2.1.1 Refinement regarding CM scope (ADV_FSP.5)

This Security Target requires a higher assurance level for family ADV_FSP compared to the PP [26], namely ADV_FSP.5 instead of ADV_FSP.4. The refinement in section 6.2.1.6 of the PP [26] regarding ADV_FSP.4 ad-

dresses the complete representation of the TSF, the purpose and method of use of all TSFIs, 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.
Compared to ADV_FSP.4 component ADV_FSP.5 requires a Functional Specification in a ”semi-formal style” (ADV_FSP.5.2C). In addition, 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 refinement in the PP [26] regarding ADV_FSP.4 can be applied without changes and is valid for ADV_FSP.5.

6.2.1.2 Refinement regarding Implementation Representation (ADV_IMP.2)

This Security Target requires a higher assurance level for family ADV_IMP compared to the PP [26], namely ADV_IMP.2 instead of ADV_IMP.1. The refinement in section 6.2.1.7 of the PP [26] regarding ADV_IMP.1 states that it must be checked 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 targets assurance level EAL6 augmented, which requires access to all source code of the TOE so that the above refinement is implicitly fulfilled.

6.2.1.3 Refinement regarding CM capabilities (ALC_CMC.5)

This Security Target requires a higher evaluation level for family ALC_CMC compared to the PP [26], namely ALC_CMC.5 instead of ALC_CMC.4. The refinement in section 6.2.1.4 of the PP [26] regarding ALC_CMC.4 is a clarification of the ”TOE” and the term ”configuration items”.
Since the higher level ALC_CMC.5 requires a higher assurance regarding the defined TOE and the configuration items, the refinement in the PP [26] regarding ADV_CMC.4 can be applied without changes and is valid for ADV_CMC.5.

6.2.1.4 Refinement regarding CM scope (ALC_CMS.5)

This Security Target requires a higher evaluation level for family ALC_CMS compared to the PP [26], namely ALC_CMS.5 instead of ALC_CMS.4. The refinement in section 6.2.1.3 of the PP [26] regarding ALC_CMS.4 is a clarification of the configuration item ”TOE implementation representation”.
Compared to ALC_CMS.4 component ALC_CMS.5 only adds the requirement for a new configuration item to be included in the configuration list (ALC_CMS.51C) so that the refinement in the PP [26] regarding ADV_CMS.4 can be applied without changes and is valid for ADV_CMS.5.

6.2.1.5 Refinements regarding Test Coverage (ATE_COV.3)

This Security Target requires a higher evaluation level for family ALC_COV compared to the PP [26], namely ATE_COV.3 instead of ATE_COV.2. The refinement in section 6.2.1.8 of the PP [26] regarding ATE_COV.2 defines that test coverage must include different operating conditions and ”ageing” and that existence and effectiveness of countermeasures against physical attacks cannot be tested but must be given by evidence.

The refinement regarding test coverage is not a change in the wording of the action elements, but a more detailed definition of the items to be applied, so that it can be applied without changes and is valid for ATE_COV.3. The refinement regarding existence and effectiveness of countermeasures against physical attacks is implicitly fulfilled since his Security Target targets assurance level EAL6 augmented, which requires access to all source code and layout data.

6.3 Security Requirements Rationale

6.3.1 Rationale for the Security Functional Requirements

Section 6.3.1 in [26] provides a rationale for the mapping between security functional requirements and security objectives defined in the PP [26]. The mapping is reproduced in the following table. Notice, that only TOE objectives are listed since no SFRs are mapped to objectives related to operational resp. development environment.

SO	SFR
O.Leak-Inherent	FDP_ITT.1[HW] FDP_IFC.1 FPT_ITT.1[HW]
O.Phys-Probing	FDP_SDC.1[HW] FPT_PHP.3
O.Malfunction	FPT_FLS.1 FRU_FLT.2
O.Phys-Manipulation	FDP_SDI.2[HW] FPT_PHP.3
O.Leak-Forced	FDP_ITT.1[HW] FDP_IFC.1 FPT_FLS.1 FPT_ITT.1[HW] FPT_PHP.3 FRU_FLT.2
O.Abuse-Func	FDP_ITT.1[HW] FDP_IFC.1 FMT_LIM.1[HW] FMT_LIM.2[HW] FPT_FLS.1 FPT_ITT.1[HW] FPT_PHP.3 FRU_FLT.2
O.Identification	FAU_SAS.1[HW]
O.RND	FCS_RNG.1[HW] FDP_ITT.1[HW]

SO	SFR
	FDP_IFC.1 FPT_FLS.1 FPT_ITT.1[HW] FPT_PHP.3 FRU_FLT.2 FCS_RNG.1[HDT] FCS_RNG.1[HPH]
O.Cap_Avail_Loader	FMT_LIM.1[Loader] FMT_LIM.2[Loader]
O.Ctrl_Auth_Loader	FDP_ACC.1[Loader] FDP_ACF.1[Loader] FDP_UCT.1 FDP_UIT.1 FTP_ITC.1
O.TDES	FCS_COP.1[TDES_HW] FCS_COP.1[TDES_SW] FCS_CKM.4[TDES_SW]
O.AES	FCS_COP.1[AES_HW] FCS_COP.1[AES_SW] FCS_CKM.4[AES_SW]

Tab. 6.48: Security Functional Requirements vs. Security Objectives (PP)

The Security Target additionally defines the SFRs for the TOE that are listed in Table 6.49. In addition Security Requirements for the Environment are defined. The following table gives an overview, how the requirements are combined to meet the security objectives.

SO	SFR
O.CUST_RECONFIG	FMT_SMF.1[HW]
O.NVM_INTEGRITY	FDP_SDI.2[HW]
O.MEM_ACCESS	FDP_ACC.1[MEM] FMT_MSA.3[MEM] FMT_MSA.1[MEM] FMT_SMF.1[HW] FDP_ACF.1[MEM]
O.SFR_ACCESS	FDP_ACC.1[SFR] FMT_MSA.3[SFR] FMT_MSA.1[SFR] FMT_SMF.1[HW] FDP_ACF.1[SFR]

SO	SFR
O.REUSE	FDP_RIP.1[SW]
O.Self-Test	FPT_TST.1
O.PUF	FCS_COP.1[AES_PUF] FCS_COP.1[MAC_PUF] FCS_CKM.1[PUF] FCS_CKM.4[PUF]
O.Reset	FMT_SMF.1[SW]
O.Secure-UM-Box-FW	FMT_SMF.1[HW] FDP_ACC.1[SUB] FDP_ACF.1[SUB] FMT_MSA.1[SUB] FMT_MSA.3[SUB]

Tab. 6.49: Security Functional Requirements vs. Security Objectives (ST)

The rationale for all items defined in the Security Target is given below.
Justification related to O.CUST_RECONFIG:

SFR	Rationale
FMT_SMF.1[HW]	This SFR is used for the specification of the management functions to be provided by the TOE as demanded by the objective.

Justification related to O.NVM_INTEGRITY:

SFR	Rationale
FDP_SDI.2[HW]	This SFR requires a control function, that adjusts the conditions of a NVM block so that integrity of the data read from it can be ensured by the TOE.

Justification related to O.MEM_ACCESS:

SFR	Rationale
FDP_ACC.1[MEM]	This SFR with the related SFP ”Hardware Access Control Policy” exactly requires to implement a memory partition as demanded by the objective.

SFR	Rationale
FMT_MSA.3[MEM]	This SFR requires that the TOE provides default values for the security attributes. 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 SFR meets the objective.
FMT_MSA.1[MEM]	This SFR requires that the ability to update the security attributes is restricted to privileged subject(s). These management functions ensure that the required access control can be realized using the functions provided by the TOE. Therefore this SFR meets the objective.
FMT_SMF.1[HW]	This SFR is used for the specification of the management functions to be provided by the TOE as demanded by the objective.
FDP_ACF.1[MEM]	This SFR with the related SFP ”Hardware Access Control Policy” defines the rules to implement the memory partition as demanded by the objective.

Justification related to O.SFR_ACCESS:

SFR	Rationale
FDP_ACC.1[SFR]	This SFR with the related SFP ”Hardware Access Control Policy” requires to implement access control for Special Function Register as demanded by this objective.
FMT_MSA.3[SFR]	This 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. There this SFR meets the objective.
FMT_MSA.1[SFR]	This SFR is realized in a way that – besides the definition of access rights to Special Function Registers related to hardware components in User Mode – no management of the security attributes is possible because the attributes are implemented in the hardware and cannot be changed. Thefore this SFR meets the objective.
FMT_SMF.1[HW]	This SFR is used for the specification of the management functions to be provided by the TOE as demanded by this objective.

SFR	Rationale
FDP_ACF.1[SFR]	This SFR with the related SFP ”Hardware Access Control Policy” defines the rules to implement the memory partition as demanded by the objective.

Justification related to O.REUSE:

SFR	Rationale
FDP_RIP.1[SW]	O.REUSE requires the TOE to provide procedural measures to prevent disclosure of memory contents that was used by the TOE. This is met by the SFR FDP_RIP.1[SW] which requires the Crypto Library to make unavailable all memory contents that has been used by it.

Justification related to O.Self-Test:

SFR	Rationale
FPT_TST.1	This SFR requires the TOE to provide self testing functionality to authorized users as required by the objective.

Justification related to O.PUF:

SFR	Rationale
FCS_COP.1[AES_PUF]	This security functional requirement defines the encryption and decryption of the user data using cryptographic algorithm AES.
FCS_COP.1[MAC_PUF]	This security functional requirement defines the calculation of a MAC as a PUF authentication value.
FCS_CKM.1[PUF]	This security functional requirement requires the generation of cryptographic key from the key derivation function based on the PUF block and the Random Number Generator (RNG). Since the PUF block and the Random Number Generator provide a TOE specific data to the key derivation function, the user data which is encrypted with this cryptographic key can be decrypted with the same TOE that the user data was encrypted on.
FCS_CKM.4[PUF]	This security functional requirement requires that the cryptographic keys which are derived by the key derivation function are destroyed by the method of flushing of key registers.

Justification related to O.Reset:

SFR	Rationale
FMT_SMF.1[SW]	This SFR requires to provide management functions allowing to reset the TOE as required by the objective.

Justification related to O.Secure-UM-Box-FW:

SFR	Rationale
FMT_SMF.1[HW]	This SFR is used for the specification of the management functions to be provided by the TOE as demanded by the objective.
FDP_ACC.1[SUB]	This SFR with the related SFP "NXP Secure User Mode Box Policy" exactly requires to implement a memory partition as demanded by the objective.
FDP_ACF.1[SUB]	This SFR with the related SFP "NXP Secure User Mode Box Policy" defines the rules to implement the memory partition as demanded by the objective.
FMT_MSA.1[SUB]	This SFR requires that the ability to update the security attributes is restricted to privileged subject(s). These management functions ensure that the required access control can be realized using the functions provided by the TOE. Therefore this SFR meets the objective.
FMT_MSA.3[SUB]	This SFR requires that the TOE provides default values for the security attributes. 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 SFR meets the objective.

6.3.2 Dependencies of Security Functional Requirements

The dependencies listed in the PP [26] are independent of the additional dependencies listed in the table below. The dependencies of the PP [26] are fulfilled within the PP [26] 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 for the requirements specified in sections 6.1 and 6.2 are satisfied.

The dependencies defined in the Common Criteria are listed in the table below:

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FAU_SAS.1[HW]	No dependencies.	No dependency

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FCS_COP.1[TDES_HW]	[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	See discussion in the PP
FCS_COP.1[TDES_SW]	[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	See discussion in the PP
FCS_COP.1[AES_HW]	[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	See discussion in the PP
FCS_COP.1[AES_SW]	[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	See discussion in the PP
FCS_CKM.4[TDES_SW]	[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]	See discussion in the PP
FCS_CKM.4[AES_SW]	[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]	See discussion in the PP
FCS_RNG.1[HW]	No dependencies.	No dependency

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FCS_RNG.1[HDT]	No dependencies.	No dependency
FCS_RNG.1[HPH]	No dependencies.	No dependency
FDP_ACC.1[Loader]	FDP_ACF.1 Security attribute based access control.	Yes
FDP_ACF.1[Loader]	FMT_MSA.3 Static attribute initialisation.	See discussion in the PP
FDP_ITT.1[HW]	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Yes
FDP_IFC.1	FDP_IFF.1 Simple security attributes	See discussion in the PP
FDP_UCT.1	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Yes
FDP_UIT.1	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	Yes
FDP_SDC.1[HW]	No dependencies.	No dependency
FDP_SDI.2[HW]	No dependencies.	No dependency
FMT_LIM.1[HW]	FMT_LIM.2 Limited availability.	Yes
FMT_LIM.1[Loader]	FMT_LIM.2 Limited availability.	Yes
FMT_LIM.2[HW]	FMT_LIM.1 Limited capabilities.	Yes
FMT_LIM.2[Loader]	FMT_LIM.1 Limited capabilities.	Yes
FPT_FLS.1	No dependencies.	No dependency
FPT_ITT.1[HW]	No dependencies.	No dependency
FPT_PHP.3	No dependencies.	No dependency
FRU_FLT.2	FPT_FLS.1 Failure with preservation of secure state.	Yes
FTP_ITC.1	No dependencies.	No dependency

Tab. 6.59: Dependencies of Security Functional Requirements (PP)

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FCS_COP.1[AES_PUF]	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction.	FCS_CKM.1[PUF] FCS_CKM.4[PUF]
FCS_COP.1[MAC_PUF]	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction.	FCS_CKM.1[PUF] FCS_CKM.4[PUF]
FCS_CKM.1[PUF]	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1[AES_PUF] FCS_CKM.4[PUF]
FCS_CKM.4[PUF]	[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.1[PUF]
FDP_ACC.1[MEM]	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1[MEM].
FDP_ACC.1[SFR]	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1[SFR].
FDP_ACC.1[SUB]	FDP_ACF.1 Security attribute based access control.	FDP_ACF.1[MEM].
FDP_ACF.1[MEM]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	FDP_ACC.1[MEM], FMT_MSA.3[MEM]
FDP_ACF.1[SFR]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	FDP_ACC.1[SFR], FMT_MSA.3[SFR]
FDP_ACF.1[SUB]	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization	FDP_ACC.1[SUB], FMT_MSA.3[SUB]

SFR	Dependencies	Fulfilled by Security Requirements in the ST
FDP_RIP.1[SW]	No dependencies.	No dependency
FMT_MSA.1[MEM]	[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	FDP_ACC.1[MEM], FMT_SMF.1[HW]. For FMT_SMR.1, see discussion below.
FMT_MSA.1[SFR]	[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	FDP_ACC.1[SFR], FMT_SMF.1[HW]. For FMT_SMR.1, see discussion below.
FMT_MSA.1[SUB]	[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	FDP_ACC.1[SFR], FMT_SMF.1[HW]. For FMT_SMR.1, see discussion below.
FMT_MSA.3[MEM]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[MEM]. For FMT_SMR.1, see discussion below.
FMT_MSA.3[SFR]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[SFR]. For FMT_SMR.1, see discussion below.
FMT_MSA.3[SUB]	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles	FMT_MSA.1[MEM]. For FMT_SMR.1, see discussion below.
FMT_SMF.1[HW]	No dependencies.	No dependency
FMT_SMF.1[SW]	No dependencies.	No dependency
FPT_TST.1	No dependencies.	No dependency

Tab. 6.60: Dependencies of Security Functional Requirements (ST)

The developer of the Security IC Embedded Software must ensure that the additional security functional requirements FCS_COP.1[AES_PUF] and FCS_COP.1[MAC_PUF] are used as specified and that the User Data processed by the related security functionality is protected as defined for the application context.
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 three components FMT_MSA.1[MEM], FMT_MSA.1[SFR] re-

spectively FMT_MSA.1[SUB] and FMT_MSA.3[MEM], FMT_MSA.3[SFR] respectively FMT_MSA.3[SUB] is not applicable within the context of the SFR. No additional definition is required, as all necessary roles are already realized via the modes of the Memory Management Unit. Furthermore, no actions by the Security IC Embedded Software are required to implement those roles. In conclusion, these dependencies are not applicable.

6.3.3 Rationale for the Assurance Requirements

The selection of assurance components is based on the underlying PP [26]. The Security Target uses the same augmentations as the PP (and the addition of augmentations ASE_TSS.2 and ALC_FLR.1), 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 [26] to choose at least EAL4 is fulfilled.
The rationale for the PP augmentations is the same as in the PP. The assurance level EAL6 is an elaborated pre-defined level of the CC, part 3 [5]. 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 EAL6. Therefore, these components add additional assurance to EAL6, but the mutual support of the requirements is still guaranteed.
As stated in the section 6.3.3 of [26], it has to be assumed that attackers with high attack potential try to attack smart cards used for digital signature applications or payment systems. Therefore specifically AVA_VAN.5 was chosen by the PP [26] 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 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[AES_PUF], FCS_COP.1[MAC_PUF] and FDP_ACC.1[MEM], FDP_ACC.1[SFR], FDP_ACC.1[SUB] with reference to the Access Control Policies defined in FDP_ACF.1[MEM], FDP_ACF.1[SFR] and FDP_ACF.1[SUB]. Therefore, these security functional requirements support the secure implementation and operation of FCS_COP.1[AES_PUF], FCS_COP.1[MAC_PUF] and of FDP_ACC.1[MEM], FDP_ACC.1[SFR] resp. FDP_ACC.1[SUB] with FDP_ACF.1[MEM], FDP_ACF.1[SFR] resp. FDP_ACF.1[SUB] as well as the dependent security functional requirements.
A Security IC hardware platform requires Security IC Embedded Software to build a secure product. Thereby the Security IC Embedded Software must support the security functionality of the hardware and implement a

sufficient management of the security services implemented in the hardware. The realization of the Security Functional Requirements within the TOE provides a good balance between flexible configuration and restrictions to ensure a secure behavior of the TOE.

7 TOE Summary Specification

7.1 Portions of the TOE Security Functionality

The TOE Security Functionality (TSF) directly corresponds to the TOE security functional requirements defined in Section 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 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 PKCC for asymmetric cryptographic algorithms.

7.1.1 Security Services

Tables 7.1 (for PP) and 7.2 (for ST) list the Security Services defined for the TOE.

Name	Title
SS.RNG	Random Number Generation
SS.HW_TDES	Triple-DES coprocessor
SS.SW_DES	Triple-DES Software Support
SS.HW_AES	AES coprocessor
SS.SW_AES	AES Software Support
SS.SW_RNG	Hybrid Deterministic/Hybrid Physical Random Number Generator
SS.Loader	Loader

Tab. 7.1: Security Services defined in the scope of the Protection Profile

Name	Title
SS.SELF_TEST	Self Test
SS.RESET	Reset Functionality
SS.RECONFIG	Post Delivery Configuration

Tab. 7.2: Security Services defined in the extended scope of this Security Target

SS.RNG **Random Number Generation**

The Random Number Generator continously 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 test functionality, which can be used by the Security IC Embedded Software to detect hardware defects or bad quality of the produced random numbers.
The TOE internally fulfils AIS31 class PTG.2 [2]. This means that the Random Number Generator also performs an online test as defined in [2] on random numbers generated for internal purposes. The Security IC Embedded Software will need to implement its own online test or use the test functionality provided by the optional Symmetric Crypto Library.

SS.HW_TDES **Triple-DES coprocessor**

The TOE provides the Triple Data Encryption Standard (Triple-DES) according to the Data Encryption Standard (DES) [12]. SS.HW_TDES is a modular basic cryptographic function, which provides the Triple-DES algorithm as defined by NIST SP 800-67 [12] by means of a hardware coprocessor and supports the 3-key Triple-DES algorithm according to keying option 1 in NIST SP 800-67 [12]. The three 56-bit keys (168-bit) for the 3-key Triple-DES algorithm shall be provided by the Security IC Embedded Software.

SS.SW_DES **Triple-DES Software Support**

SS.SW_DES supports additional modes of operation on top of SS.HW_TDES. The supported modes are ECB, CBC, Retail-MAC, and CMAC (i.e. the CBC mode applied to the block cipher algorithm 3DES). In addition, the TOE provides the ability to compute a CBC-MAC. The CBC-MAC mode of operation is rather similar to the CBC mode of operation, but returns only the last cipher text. Like ECB and CBC, the CBC-MAC mode of operation can also be applied to 3DES as underlying block cipher algorithm.

The interface to SS.SW_DES allows performing 3-key Triple-DES operations independent from prior key loading. The user has to take care that adequate keys of the correct size are loaded before the cryptographic operation is performed. Details are described in the user manual.

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 [8]. 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.SW_AES **AES Software Support**

The TOE uses the AES hardware coprocessor to provide AES encryption and decryption facility using 128, 192 or 256 bit keys. The TOE implements the AES with different security configurations. The supported modes are ECB, ”outer” CBC and CMAC (i.e. the CBC mode applied to the block cipher algorithm AES). In addition, the TOE provides the ability to compute a CBC-MAC. The CBC-MAC mode of operation is rather similar to the CBC mode of operation, but returns only the last cipher text.

The interface to SS.SW_AES allows AES operations independent from prior key loading. The user has to take care that adequate keys of the correct size are loaded before the cryptographic operation is performed. Details are described in the user guidance and the user manual.

SS.SW_RNG **Hybrid Deterministic/Hybrid Physical Random Number Generator**

The TOE implements a software (pseudo) RNG that can be used as a general purpose random source. This software RNG has to be seeded by random numbers taken from the hardware RNG provided via SS.RNG. The implementation of the software RNG is based on the standard NIST-SP-800-90 CTR-DBRG.

SS.Loader **Loader**

SS.Loader is implemented by the Flashloader OS running after personalization in the production environment typically at the customer environment (or at a third party personalizer) before the product is shipped to the end customer for usage. It allows to configure the device and especially supports the download of code and data in a secure way to flash memory. The flashloader protocol ensures by means of authentication that only authorized parties are able to configure the card and to download code/data. Furthermore, it ensures that integrity and confidentiality is preserved. Once the flashloader is terminated after personalization it cannot be activated anymore in the system. An update of code and/or data in the field at the end-customer is not foreseen.

SS.SELF_TEST **Self Test**

SS.SELF_TEST allows checking whether the TOE has been physically manipulated. This includes an active shielding check, sensor check, verifying the signature of code and performing a consistency check of special-function registers with static configuration.

SS.RESET **Reset Functionality**

SS.RESET provides the Security IC Embedded Software with a function to reset the device. This enables the Security IC Embedded Software preserving a secure state in case it detects abnormal operations or attacks. The reset functionality provides an ordinary System Reset (that is, ”Power-On Reset”) and a security relevant reset (Security Reset) which can be executed only a limited time before the device is disabled permanently by the Security IC Embedded Software. The IC can also be terminated with one call, where the error counter is set to its end state.

SS.RECONFIG **Post Delivery Configuration**

SS.RECONFIG realizes the Post Delivery Configuration. This allows the customer to configure the device after delivery by NXP. These configuration options include commercial settings like disabling the AES, DES and PKCC co-processor, reducing the size of available Flash memory, setting the operation mode and the number of logical cards. Furthermore, functional settings like wear-level partition of each logical card, Flash partition of each logical card, the free page pool size, and code bases for system mode A and B can be modified. The changing of settings is done in a secure way using a diversified password scheme.

The customer can change these values as many times as he wishes. However, once he calls the Boot 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

Tables 7.3 (for PP) and 7.4 (for ST) list the Security Features defined for the TOE.

Name	Title
SF.OPC	Control of Operating Conditions
SF.PHY	Protection against Physical Manipulation
SF.LOG	Logical Protection
SF.COMP	Protection of Mode Control

Tab. 7.3: Security Features defined in the scope of the Protection Profile

Name	Title
SF.MEM_ACC	Memory Access Control
SF.SFR_ACC	Special Function Register Access Control
SF.MEM_SUB	Secure User Mode Box Firewall
SF.Object_Reuse	Reuse of Memory
SF.PUF	PUF

Tab. 7.4: Security Features defined in the extended scope of this Security Target

SF.OPC **Control of Operating Conditions**

SF.OPC ensures the correct operation of the TOE (functions offered by the micro-controller including the standard CPU as well as the unified AES/Triple-DES co-processor, the memories, registers, I/O interfaces and the other system peripherals) during the execution of the IC Ded-

icated Support Software and the Security IC Embedded Software. This includes all specific security features of the TOE which are able to provide an active response.

The TOE ensures its correct operation and prevents any malfunction my means of three kinds of features:

Environmental Control: Set of security mechanisms that detect if the TOE runs out of the specified operation conditions. It needs to be assured that in operation mode all ambient conditions are within their specified limits. Sensors take over the role of measuring the ambient conditions and reacting in case of specification violation of one of the ambient parameters. If a sensors triggers a reset is triggered. Depending on the type of sensor the reset might be a security reset that increments the error counter.

Execution Integrity Set of security mechanisms that detect if an execution of an operation has been manipulated. It needs to be assured that manipulations on operations are detected and trigger a reset that effects the error counter. Manipulating operations means the operation itself is attacked. On an abstract view this could mean that some kind of memory (e.g. register) has been attacked. On a more detailed view it can also mean that entire wires or gates are attacked. Executing integrity is achieved by means such as the following ones:

- validity checking of in- and output of security critical operations
- integrity protection of data, code and address path
- integrity protection of memories, data registers, key registers and control registers
- monitoring state machines
- integrity protection of sensor signals
- double calculations and checks

Integrity protection is achieved by various techniques, such as parity protection, redundant encoding and execution, monitoring, CRCs.

Availability Set of security mechanisms that take care that the availability of the TOEs functionality is limited if attacks occur. It needs to be assured that the detection of an attack results in secure state. This is achieved by the fact that any kind of attack or operation outside the operation conditions results in a reset. Depending in the kind of reset source the reset might also have an effect on the error counter. This is especially the case for integrity violations that cannot be unintended ones.

SF.PHY **Protection against Physical Manipulation**

The feature SF.PHY protects the TOE against manipulation of

(i) the hardware,

(ii) the IC Dedicated Software in the ROM or Flash,

(iii) the Security IC Embedded Software in the ROM or Flash and

(iv) the application data in the RAM and Flash including the configuration data stored in Flash.

It also protects all data stored in the memories including User Data and TSF data against disclosure by physical probing when stored or while being processed by the TOE.

The TOE ensures its correct operation and prevents any malfunction by means of several kinds of features:

- **Layout Protection:** Set of security mechanisms that hamper reverse engineering of the IC, such as layout randomization, active and passive shielding, techniques to hide shielding, multilayer interconnection, wide bus widths and dummy routing.
- **Code- & Datapath Integrity Protection:** Set of security mechanisms that ensure that manipulations on data or code stored and transmitted from resp. to the CPU are detected with high probability. This includes integrity protection of the whole code and data path including CPU internals. Integrity verification is always done before the according data is processed via e.g. an ALU operation.
- **Memory Integrity Protection:** Set of security mechanisms that ensure that manipulations on memory content are detected with high probability. This includes integrity protection of memories and registers. Flash are additionally equipped with error correction codes, double read technology and anti-tearing.
- **Address Path Integrity Protection:** Set of security mechanisms that ensure that manipulations on the address path are detected with high probability.
- **Startup Integrity Protection:** Set of security mechanisms that detect integrity errors during startup (e.g. w.r.t. configuration data).
- **Redundant Encoding:** Set of security mechanims that ensure that security critical flags and the according checks are kept with an according level of redundancy.
- **Code Integrity Protection:** Set of security mechanisms that detect if code has been manipulated. This is especially checked by SS.SELF_TEST.
- **Code- & Datapath Encryption:** Set of security mechanisms that ensure that code or data processed by the CPU is stored and transmitted in encrypted form. All data transmitted over the code or datapath is encrypted with an address-dependent non-linear encryption scheme. En- and decryptions are performed in the CPU core.
- **Address Scrambling:** Set of security mechanisms that ensure that physical addresses are scrambled before writing data to the memory.
- **Code- & Datapath Key Management:** Set of security mechanisms that ensure that keys used for the secure data path are derived correctly and securely. This includes address dependent key derivation functionality with an according strength of diffusion and confusion to achieve a good avalanche effect.

SF.LOG **Logical Protection**

SF.LOG implements measures to limit or eliminate the information that might be contained in the shape and amplitude of signals or in the time between events found by measuring such signals. This comprises the power consumption and signals on the other pads that are not intended by the terminal or the Security IC Embedded Software. Thereby SF.LOG prevents the disclosure of User Data or TSF data stored and/or processed in the security IC through the measurement of the power consumption or emanation and subsequent complex signal processing. The protection of the TOE comprises different features within the design that support the other portions of security functionality.

The cryptographic co-processor includes special features to hamper SPA/DPA analysis of shape and amplitude of the power consumption and ensures that the calculation time is independent from any key and plain/cipher text. These include blinding and randomization techniques.
Specific features as described for SF.PHY (e.g. the encryption features) and for SF.OPC (e.g. the filter feature) support the logical protection. For instance, the encryption of the whole data and code path including memory and register contents.

SF.COMP **Protection of Mode Control**

SF.COMP provides a control of the TOE modes. This includes the protection of electronic fuses stored in a protected memory area, and the possibility to store initialisation or pre-personalisation data in the so-called FabKey Area.

The control of the TOE modes prevent the abuse of test functions after TOE delivery. Additionally it also ensures that features used during the boot sequence to configure the TOE can not be abused. Hardware circuitry and the Boot Software determine whether the test functionality is available or not. If it is available, the TOE starts the IC Dedicated Test Software in Super System Mode (Test Mode). Otherwise, the TOE switches to the User Mode or System Mode and starts execution of the Security IC Embedded Software.
The switch to the IC Dedicated Test Software is prevented after TOE delivery because specific electronic fuses guarantee that the IC Dedicated Test Software cannot be selected. The System Mode is the more privileged TOE mode, the User Mode is the less privileged TOE mode. The Boot Software is executed in Super System Mode. For the Security IC Embedded Software, User Mode and System Mode are available.
The protection of the electronic fuses especially ensures that configuration options with regard to the security functionality cannot be changed, abused or influenced in any way in System Mode and User Mode. SF.COMP ensures that activation or deactivation of security features cannot be influenced by the Security IC Embedded Software.
SF.COMP limits the capabilities of the test functions and provides test personnel during phase 3 with the capability to store the identification and/or pre-personalization data in the Flash.

SF.MEM_ACC **Memory Access Control**

SF.MEM_ACC ensures that access to all logical and physical addresses is properly controlled by the TOE. The access control decisions are based on the current active mode and currently active logical card of the TOE as well as additional attributes that might be set for certain memory areas.

SF.SFR_ACC **Special Function Register Access Control**

SF.SFR_ACC ensures that access to all special-function registers is properly controlled by the TOE. The access control decisions are based on the current active mode and currently active logical card of the TOE as well as whether a peripheral is already owned by another logical card or mode. Furthermore, address ranges and read/write restrictions are controlled.

SF.MEM_SUB **Secure User Mode Box Firewall**

SF.MEM_SUB (NXP secure user mode box firewall) ensures that whatever user mode code is executed on top of the by NXP provided and configured System Mode OS in logical card A cannot endanger any asset of the TOE. This security feature allows to change the user mode code in logical card A (e.g. a MIFARE emulation) without any impact on the base certificate but also without any impact on the operating system running in logical card B. The protection is ensured by the access control policy of the memory system and the special-function register access control policy.

SF.Object_Reuse **Reuse of Memory**

SF.Object_Reuse provides internal security measures which clear memory areas used by the Crypto Library after usage. These measures ensure that a subsequent process may not gain access to cryptographic assets stored temporarily in memory used by the TOE.

SF.PUF **PUF**

SF.PUF implements a mechanism to seal/unseal user data stored in shared memory against unintended disclosure. SF.PUF encrypts/decrypts user data with a cryptographic key derived from the PUF data. SF.PUF calculates a MAC as a PUF authentication value. SF.PUF provides this sealing/unsealing mechanism through the interfaces of the IC Dedicated Software only. The AES functionality of the TOE is mandatory for this feature.

8 Bibliography

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

[2] Anwendungshinweise und Interpretationen zum Schema, AIS31: Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren, Bundesamt für Sicherheit in der Informationstechnik. Version 2.0, September 18, 2011.

[3] Common Criteria for Information Technology Security Evaluation, Part 1 – Introduction and general model - Version 3.1 CCMB-2012-09-001, Revision 4, September 2012.

[4] Common Criteria for Information Technology Security Evaluation, Part 2 – Security functional components, Version 3.1 CCMB-2012-09-002, Revision 4, September 2012.

[5] Common Criteria for Information Technology Security Evaluation, Part 3 – Security assurance components, Version 3.1 CCMB-2012-09-003, Revision 4, September 2012.

[6] Common Methodology for Information Technology Security Evaluation CEM-99/045 Part 2 - Evaluation Methodology, Version 3.1 CCMB-2012-09-004, Revision 4, September 2012.

[7] Crypto Library Iron on N7021 VA, Information on Guidance and Operation, NXP Semiconductors, Document number 3300**.

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

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

[10] NIST Special Publication 800-38A Recommendation for BlockCipher Modes of Operation. http://csrc.nist.gov/publications/nistpubs/800-38a/sp800-38a.pdf.

[11] NIST Special Publication 800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf.

[12] NIST Special Publication 800-67 Revision 2 Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher. https://doi.org/10.6028/NIST.SP.800-67r2.

[13] NXP Secure Smart Card Controller N7021, Information on Guidance and Operation, NXP Semiconductors, Document number 3318**.

[14] Objective data sheet addendum SmartMX3 N7021, Chip Health Mode, NXP Semiconductors, Document number 3298**.

[15] Objective data sheet addendum SmartMX3 N7021, Flash Loader, including guidance and operation, NXP Semiconductors, Document number 3309**.

[16] Objective data sheet addendum SmartMX3 N7021, Instruction Set Manual, NXP Semiconductors, Document number 3290**.

[17] Objective data sheet addendum SmartMX3 N7021, Inter-Card Communication Functionality and additional APIs, NXP Semiconductors, Document number 3444**.

[18] Objective data sheet addendum SmartMX3 N7021, MMU configuration and firmware interface, NXP Semiconductors, Document number 3320**.

[19] Objective data sheet addendum SmartMX3 N7021, NVM Operate Function, NXP Semiconductors, Document number 3704**.

[20] Objective data sheet addendum SmartMX3 N7021, Peripheral configuration and use, NXP Semiconductors, Document number 3321**.

[21] Objective data sheet addendum SmartMX3 N7021, Post Delivery Configuration, NXP Semiconductors, Document number 3299**.

[22] Objective data sheet addendum SmartMX3 N7021, Shared OS libraries, NXP Semiconductors, Document number 3314**.

[23] Objective data sheet addendum SmartMX3 N7021, System Mode OS interface, NXP Semiconductors, Document number 3317**.

[24] Objective data sheet addendum SmartMX3 N7021, Wafer and delivery specification, NXP Semiconductors, Document number 3313**.

[25] Objective data sheet SmartMX3 family P71D320, Overview, pinning and electrical characteristics, NXP Semiconductors, Document number 3295**.

[26] Security IC Platform Protection Profile with Augmentation Packages, registered and certified by Bundesamt fuer Sicherheit in der Informationstechnik (BSI) under the reference BSI-CC-PP-0084-2014, Rev 1.0, 13 January 2014.

[27] User Manual N7021 Crypto Library, RNG Library, NXP Semiconductors, Document number 3304**.

[28] User Manual N7021 Crypto Library, Symmetric Cipher Library (SymCfg), NXP Semiconductors, Document number 3302**.

[29] User Manual N7021 Crypto Library, Utils Library, NXP Semiconductors, Document number 3301**.

[30] ISO/IEC 14443-3: Identification cards, Contactless integrated circuit(s) cards - Proximity cards, Part 3: Initialization and anticollision, 11 2009.

[31] ISO/IEC 9797-1: Information technology - Security techniques - Message Authentication Codes (MACs) - Part 1: Mechanisms using a block cipher, 2011.

[32] PUF Key derivation function specification, NXP Semiconductors, Business Unit Identification (draft), 2014.

9 Contents

1 ST Introduction **2**
1.1 ST Reference . . . 2
1.2 TOE Reference . . . 2
1.3 TOE Overview . . . 2
1.3.1 Usage and Major Security Functionality of the TOE . . . 2
1.3.2 TOE Type . . . 4
1.3.3 Required non-TOE Hardware/Software/-Firmware . . . 4
1.4 TOE Description . . . 4
1.4.1 Physical Scope of TOE . . . 4
1.4.2 Evaluated Configurations . . . 7
1.4.3 Logical Scope of TOE . . . 11
1.4.4 Security during Development and Production . . . 15
1.4.5 TOE Intended Usage . . . 16
1.4.6 Interface of the TOE . . . 17

2 Conformance Claims **18**
2.1 Package Claim . . . 18
2.2 PP Claim . . . 18
2.3 Conformance Claim Rationale . . . 19

3 Security Problem Definition **20**
3.1 Description of Assets . . . 20
3.2 Threats . . . 20
3.3 Organizational Security Policies . . . 23
3.4 Assumptions . . . 24

4 Security Objectives **26**
4.1 Security Objectives for the TOE . . . 26
4.2 Security Objectives for the Security IC Embedded Software Development Environment 30
4.3 Security Objectives for the Operational Environment . . . 31
4.4 Security Objectives Rationale . . . 32

5 Extended Components Definitions **37**

6 Security Requirements **38**
6.1 Security Functional Requirements . . . 38
6.1.1 SFRs of the Protection Profile . . . 40
6.1.2 Additional SFRs regarding Cryptographic Support . . . 52
6.1.3 Additional SFRs regarding Protection of TSF . . . 53
6.1.4 Additional SFRs regarding Security Management . . . 53
6.1.5 Additional SFRs regarding User Data Protection . . . 54
6.1.6 Additional SFRs regarding Access Control 54
6.2 Security Assurance Requirements . . . 66
6.2.1 Refinements of the TOE Security Assurance Requirements . . . 68
6.3 Security Requirements Rationale . . . 70
6.3.1 Rationale for the Security Functional Requirements . . . 70
6.3.2 Dependencies of Security Functional Requirements . . . 75
6.3.3 Rationale for the Assurance Requirements 80
6.3.4 Security Requirements are Internally Consistent . . . 80

7 TOE Summary Specification **82**
7.1 Portions of the TOE Security Functionality . 82
7.1.1 Security Services . . . 82
7.1.2 Security Features . . . 85

8 Bibliography **90**

9 Contents **93**

10 Legal information **95**
10.1 Definitions 95
10.2 Disclaimers 95
10.3 Licenses 95
10.4 Patents 95
10.5 Trademarks 95

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 life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an 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.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP 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 competent authorities.

Evaluation products – This product is provided on an "as is" and "with all faults" basis for evaluation purposes only. NXP Semiconductors, its affiliates and their suppliers expressly disclaim all warranties, whether express, implied or statutory, including but not limited to the implied warranties of non-infringement, merchantability and fitness for a particular purpose. The entire risk as to the quality, or arising out of the use or performance, of this product remains with customer.

In no event shall NXP Semiconductors, its affiliates or their suppliers be liable to customer for any special, indirect, consequential, punitive or incidental damages (including without limitation damages for loss of business, business interruption, loss of use, loss of data or information, and the like) arising out the use of or inability to use the product, whether or not based on tort (including negligence), strict liability, breach of contract, breach of warranty or any other theory, even if advised of the possibility of such damages.

Notwithstanding any damages that customer might incur for any reason whatsoever (including without limitation, all damages referenced above and all direct or general damages), the entire liability of NXP Semiconductors, its affiliates and their suppliers and customer's exclusive remedy for all of the foregoing shall be limited to actual damages incurred by customer based on reasonable reliance up to the greater of the amount actually paid by customer for the product or five dollars (US$5.00). The foregoing limitations, exclusions and disclaimers shall apply to the maximum extent permitted by applicable law, even if any remedy fails of its essential purpose.

10.3 Licenses

ICs with DPA Countermeasures functionality

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

10.4 Patents

Notice is herewith given that the subject device uses one or more of the following patents and that each of these patents may have corresponding patents in other jurisdictions.
<Patent ID> – owned by <Company name>

10.5 Trademarks

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

MIFARE – is a trademark of NXP N.V.

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

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

Date of release: 2019-06-04

Document identifier: BSI-DSZ-CC-0977