



**PUBLIC**

**Infineon Technologies AG**  
**Chipcard and Security**  
**Evaluation Documentation**  
**Security Target**  
**Assurance Continuity Maintenance**

**M7820 A11**

**including optional Software Libraries**  
**RSA - EC - SHA-2 - Toolbox**

**Version** 2.2  
**Date** 2017-10-19  
**Author** Hans-Ulrich Buchmüller

---

Filename: M7820\_ACM\_Security\_Target

---

## REVISION HISTORY

2.1	2017-09-28 The version number is taken from on the forerunner process
2.2	2017-10-19 Update of User Guidance

## TABLE OF CONTENTS

<b>1</b>	<b>SECURITY TARGET INTRODUCTION (ASE_INT)</b> .....	<b>6</b>
1.1	Security Target and Target of Evaluation Reference.....	6
1.2	Target of Evaluation overview .....	13
<b>2</b>	<b>TARGET OF EVALUATION DESCRIPTION</b> .....	<b>17</b>
2.1	TOE Definition.....	17
2.2	Scope of the TOE.....	20
2.2.1	Hardware of the TOE .....	21
2.2.2	Firmware and software of the TOE.....	22
2.2.3	Interfaces of the TOE .....	23
2.2.4	Guidance documentation .....	23
2.2.5	Forms of delivery .....	25
2.2.6	Production sites.....	26
<b>3</b>	<b>CONFORMANCE CLAIMS (ASE_CCL)</b> .....	<b>27</b>
3.1	CC Conformance Claim.....	27
3.2	PP Claim.....	27
3.3	Package Claim.....	27
3.4	Conformance Rationale .....	28
3.5	Application Notes .....	29
<b>4</b>	<b>SECURITY PROBLEM DEFINITION (ASE_SPD)</b> .....	<b>30</b>
4.1	Threats .....	30
4.1.1	Additional Threat due to TOE specific Functionality .....	30
4.1.2	Assets regarding the Threats .....	31
4.2	Organizational Security Policies .....	31
4.2.1	Augmented Organizational Security Policy .....	32
4.3	Assumptions .....	32
4.3.1	Augmented Assumptions .....	34
<b>5</b>	<b>SECURITY OBJECTIVES (ASE_OBJ)</b> .....	<b>35</b>

5.1	Security objectives for the TOE .....	35
5.2	Security Objectives for the development and operational Environment.....	36
5.2.1	Clarification of “Usage of Hardware Platform (OE.Plat-Appl)” .....	36
5.2.2	Clarification of “Treatment of User Data (OE.Resp-Appl)” .....	36
5.2.3	Clarification of “Protection during Composite product manufacturing (OE.Process-Sec-IC)” .....	37
5.3	Security Objectives Rationale .....	37
<b>6</b>	<b>EXTENDED COMPONENT DEFINITION (ASE_ECD).....</b>	<b>39</b>
6.1	Component “Subset TOE security testing (FPT_TST)” .....	39
6.2	Definition of FPT_TST.2.....	39
6.3	TSF self-test (FPT_TST) .....	40
6.4	Family “Generation of Random Numbers (FCS_RNG)” .....	40
6.5	Definition of FCS_RNG.1 .....	41
<b>7</b>	<b>SECURITY REQUIREMENTS (ASE_REQ) .....</b>	<b>42</b>
7.1	TOE Security Functional Requirements .....	42
7.1.1	Extended Components FCS_RNG.1 and FAU_SAS.1 .....	44
7.1.2	Subset of TOE testing.....	45
7.1.3	Memory access control .....	45
7.1.4	Support of Cipher Schemes .....	48
7.1.5	Data Integrity.....	57
7.2	TOE Security Assurance Requirements.....	58
7.2.1	Refinements.....	58
7.3	Security Requirements Rationale.....	59
7.3.1	Rationale for the Security Functional Requirements.....	59
7.3.2	Rationale of the Assurance Requirements .....	64
<b>8</b>	<b>TOE SUMMARY SPECIFICATION (ASE_TSS).....</b>	<b>66</b>
8.1	SF_DPM: Device Phase Management.....	66
8.2	SF_PS: Protection against Snooping .....	67
8.3	SF_PMA: Protection against Modifying Attacks .....	68
8.4	SF_PLA: Protection against Logical Attacks .....	70
8.5	SF_CS: Cryptographic Support.....	70
8.5.1	3DES.....	71
8.5.2	AES.....	71
8.5.3	RSA.....	71
8.5.4	Elliptic Curves .....	72
8.5.5	SHA-2 .....	74
8.5.6	Toolbox Library .....	74
8.5.7	Base Library .....	74
8.5.8	PTRNG respectively TRNG.....	74

---

8.6	Assignment of Security Functional Requirements to TOE's Security Functionality .....	75
8.7	Security Requirements are internally Consistent .....	77
9	REFERENCES .....	78
9.1	Literature.....	78
10	APPENDIX.....	79
11	LIST OF ABBREVIATIONS .....	81
12	GLOSSARY.....	83

**List of tables:**

Table 1: Identification	7
Table 2: Alternative Firmware Packages	8
Table 3: Options to implement user software at Infineon production premises	9
Table 4: Basic Hardware Configurations of the TOE	9
Table 5: Today's predefined hardware configurations and derivatives	11
Table 6: TOE forms and delivery	25
Table 7: Production site in chip identification	26
Table 8: Augmentations of the assurance level of the TOE	27
Table 9: Threats according PP [1]	30
Table 10: Additional threats due to TOE specific functions and augmentations	30
Table 11: Organizational Security Policies according PP [1]	31
Table 12: Assumption according PP [1]	33
Table 13: Objectives for the TOE according to PP [1]	35
Table 14: Additional objectives due to TOE specific functions and augmentations	36
Table 15: Security objectives for the environment according to PP [1]	36
Table 16: Security Objective Rationale	37
Table 17: Security functional requirements defined in PP [1]	42
Table 18: Augmented security functional requirements	43
Table 19: Cryptographic TOE Functionality	49
Table 20: Assurance components	58
Table 21: Rationale for additional SFR in the ST	59
Table 22: Dependency for cryptographic operation requirement	62
Table 23: Mapping of SFR and SF	76

## 1 Security Target Introduction (ASE\_INT)

All referenced product or service names and trademarks are the property of their respective owners.

### 1.1 Security Target and Target of Evaluation Reference

The title of this document is Security Target (ST) M7820 A11 Assurance Continuity Maintenance including optional Software Libraries RSA - EC - SHA-2 - Toolbox and comprises the Infineon Technologies Smart Card IC (Security Controller) M7820 A11 with specific IC dedicated software and optional RSA v1.02.013, EC v1.02.013, SHA-2 v1.01 and Toolbox v1.02.013 cryptographic libraries.

The target of evaluation (TOE) M7820 A11 is described in the following. The Security Target has the revision 2.2 and is dated 2017-10-19.

The Target of Evaluation (TOE) is an Infineon smart card IC (Security Controller) M7820 A11 with optional RSA2048/4096 v1.02.013, EC v1.02.013, SHA-2 v1.01 and Toolbox v1.02.013 libraries and with specific IC dedicated software. More details are listed in Table 1: Identification and its blocked derivatives listed in Table 5. The design steps of this TOE are A11.

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

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

The ST takes into account all relevant current final interpretations.

This TOE is built from the equal hardware design sources as certified in the process BSI-DSZ-CC-0728-2011, BSI-DSZ-CC-0813-2012 and BSI-DSZ-CC-0695-2011.

Just the user guidance documents have been updated.

Table 1: Identification

Object	Version	Date	Registration
Security Target	2.2	2017-10-19	M7820 A11
Target of Evaluation	A11		<p>M7820</p> <p>with Flash Loader (FL) V3.60.009 and FL Patch V3.61.006 and RMS V8000B001B and STS V78.01.09.09 and STS Patch V800B and SAM V20.22 and Mifare(1) V8002B0002 and various FW patches and Overall-Patch 8047 and optional SW:</p> <p>RSA2048 v1.02.013 (optional) RSA4096 v1.02.013 (optional) EC v1.02.013 (optional) SHA-2 v1.01 (optional) Toolboxv1.02.013 (optional) and guidance documentation</p>
Guidance Documentation	Edition	2010-11-18	SLx 70 Family – Hardware Reference Manual
		2015-04-01	SLx 70 Family Production and Personalization User’s Manual
		2017-07-04	SLE 70 Family Programmer’s Reference User’s Manual
		2013-04-17	M7801 / M7820 Controller Family for Security Applications, Errata Sheet
	v1.02.013	2011-06-07	SLE70 Asymmetric Crypto Library Crypto@2304T, RSA / ECC / Toolbox, Users Interface (optional) date of additional Errata Sheet Section
		2017-06-20	
		2009-11	Chipcard and Security ICs, SLx70 Family, Secure Hash Algorithm SHA-2, (SHA 256/224, SHA 512/384) (optional)
		2010-03-23	Crypto@2304T User Manual
		2017-09-18	M7820 Controller Family for Security Application, Security Guidelines
Protection Profile	1.0	2007-06-15	Security IC Platform Protection Profile PP0035
Common Criteria	3.1	2017-04	Common Criteria for Information Technology Security Evaluation
	Revision 5		<p>Part 1: Introduction and general model CCMB-2017-04-001</p> <p>Part 2: Security functional requirements CCMB-2017-04-002</p> <p>Part 3: Security Assurance Components CCMB-2017-04-003</p>

This TOE comes with two alternative firmware packages as given in following table:

Table 2: Alternative Firmware Packages

Firmware	Alternative Package 1	Alternative Package 2
RMS	8000.001B	8000.001B
RMS-Patch	0048	v0021
STS	78010909	78010909
STS-Patch	800B	800B
SAM	20.22	20.22
SAM Patch	0003	0003
FL70	3.60.009	3.60.009
FL Patch	3.61.006	3.61.002
Mifare	8002.0002	8002.0002
Mifare Patch	0002	0001
Overall FW Patch	8047	8014

This means that depending on the order the TOE products come either with Firmware package 1 or firmware package 2. Both firmware packages are part of the evaluation.

(1) The term “Mifare” in this document is only used as an indicator of product compatibility to the corresponding established technology. This applies to the entire document wherever the term is used.

All products based on the M7820 representing this TOE are identically from hardware perspective and produced with the same masks. The first metal mask (called M1 mask) contains the derivate specific information (e.g. development code, first digit of the design step and i.e. ROM mask data).

Depending on the blocking configuration an M7820 product can have different user available memory sizes and can come with or without individual accessible cryptographic co-processors. For example a product with the M-number M7820 in the field can come in one project with the fully available SOLID FLASH™ NVM<sup>1</sup> or in another project with equal or any other SOLID FLASH™ NVM-size below the physical implementation size, depending on the user requirements. And more, the user is free to choice prior to production, whether he needs the symmetric co-processor SCP, or the asymmetric co-processor Crypto2304T, or both, or none of them.

The entire configuration is done during the manufacturing process of the TOE according to the choice of the user. All differences between the products of this TOE are realized by means of blocking without changing the hardware. Therefore, all products of this TOE are equal from hardware perspective.

The blocking of the SOLID FLASH™ NVM is done by setting the according value in the chip configuration page, which is not available to the user. The same means of blocking are also used for switching on and off the accessibility of the cryptographic co-processors SCP and/or Crypto2304T and also for the configuration of the XRAM- and ROM-sizes.

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

This TOE is equipped with Flash Loader software (FL) to allow the download of user software, i.e. the operating system and applications. Various options can be chosen by the user to implement his software during production providing a maximum of flexibility:

---

<sup>1</sup> Nonvolatile memory equals in the meaning to the Electrical Erasable Programmable Read Only Memory (EEPROM)



Table 3: Options to implement user software at Infineon production premises

1.	The user or/and a subcontractor downloads the software into the SOLID FLASH™ NVM on his own. Infineon Technologies has not received user software and there are no user data in the ROM.	The FL can be activated or reactivated by the user or subcontractor to download his software in the SOLID FLASH™ NVM.
2	The user provides software for the download into the SOLID FLASH™ NVM to Infineon Technologies AG. The software is downloaded to the SOLID FLASH™ NVM memory during chip production. I.e. there are no user data in the ROM.	There is no FL present.
3	The user provides software for the download into the SOLID FLASH™ NVM to Infineon Technologies AG. The software is downloaded to the SOLID FLASH™ NVM during chip production. I.e. there are no user data in the ROM	The FL is blocked afterwards but can be activated or reactivated by the user or subcontractor to download his software in the SOLID FLASH™ NVM. Precondition is that the user has provided an own reactivation procedure in software prior chip production to Infineon Technologies AG.
4	The user provides the software for implementation into the ROM mask.	There is no FL present.
5	The user provides the software for implementation into the ROM mask.	The FL is blocked afterwards but can be activated or reactivated by the user or subcontractor to download his software in the SOLID FLASH™ NVM. Precondition is that the user has provided an own reactivation procedure in software prior chip production to Infineon Technologies AG.
6	The user provides the software for implementation into the ROM mask and provides software for the download into the SOLID FLASH™ NVM to Infineon Technologies.	There is no FL present.
7	The user provides the software for implementation into the ROM mask and provides software for the download into the SOLID FLASH™ NVM to Infineon Technologies.	The FL is blocked afterwards but can be activated or reactivated by the user or subcontractor to download his software in the SOLID FLASH™ NVM. Precondition is that the user has provided an own reactivation procedure in software prior chip production to Infineon Technologies AG.

For the cases with Flash Loader on board and whenever the user has finalized his SW-download, the user is obligated to lock the Flash Loader. The final locking of the FL results in a permanent deactivation of the Flash Loader. This means that once being in the locked status, the Flash Loader cannot be reactivated anymore.

Within its physical limits various configuration can occur which are and will all be equal from hardware perspective. Anyhow the user must be able to clearly identify whether a certain product is covered by a certificate or not.

The following table contains memory size regions and other blocking options within the configuration can vary under only one development code – the M7820.

The Target of Evaluation (TOE) comprises the following product:

Table 4: Basic Hardware Configurations of the TOE

TOE functionality	Configuration option
TOE memories	
SOLID FLASH™ NVM	Up to 160 kByte
ROM	Up to 280 kByte
XRAM	Up to 8 kByte

TOE functionality	Configuration option
TOE peripherals	
Symmetric Coprocessor	Accessible or blocked
Crypto@2304T coprocessor	Accessible or blocked
TOE software	
SHA library v1.01	Available or not available
Firmware Package (see table 2)	Package 1 or Package 2
Communication interfaces	
ISO 7816	Accessible or blocked
ISO 14443 Type A	Accessible or blocked
ISO 14443 Type B	Accessible or blocked
ISO/IEC 18092 passive mode	Accessible or blocked
Mifare compatible interface	Accessible or blocked
Digital Contactless Bridge (DCLB) (1)	Accessible or blocked
Advanced Communication Mode (ACM) (2)	Accessible or blocked
IC Embedded Software (see table 3)	
Location	ROM or FLASH
Flash Loader	Accessible or blocked

(1) DCLB = Digital Contactless Bridge, a contact based communication mode using a contactless protocol

(2) ACM = Advanced Communication Modes, enabling for very high bit rates according to ISO/IEC 14443-2AM3: Information Technology – Identification Cards – Contactless Integrated Circuit(s) Cards - Proximity Cards; Part 2: Radio frequency power and signal interface [2010]; AMENDMENT 3: Bits rates of  $f_c/8$ ,  $f_c/4$  and  $f_c/2$

Note that any hardware configuration can come with or without the optional software. This is defined by the user order.

Beside these flexible ranges, the user guidance contains a number of predefined products with different configurations. All of these are labelled with M7820 and are of course made of the equal hardware and belong to this TOE as well. Today's configurations of the TOE are listed below. These predefined products come with the most requested configurations and allow to produce volumes on stock in order to simplify logistic processes.

Note that any hardware configuration comes with its own chip identifier byte as shown in the table below. The chip identifier bytes are aimed to be used for simplification of the logistical processes but are available to the user as well.

For the user's clear TOE identification, the ChipIdent contains the relevant data which clearly can be mapped to a product of the TOE in a dedicated configuration. The hardware reference manual [7] allows the clear interpretation of the read out ChipIdent data.

In addition, a dedicated RMS function allows reading out the present configuration of a given M7820 derivative, which also allows for clear identification of a certain configuration with the help of the hardware reference manual [7]. The ACM value can be determined by reading a dedicated register value. It is not linked to a certain Chip Identifier Byte.

Table 5: Todays predefined hardware configurations and derivatives

No.	Product Code	Chip Identifier Byte	Sales Name	NVM(1) kBytes	ROM(1) kBytes	XRAM(1) kBytes	ISO 7816(2)	ISO 14443	Mifare compatible Interface	ISO/IEC18092 Passive mode	DCLB
1	M7820	A0h	SLE78CLX1440P	144	280	8	Yes	Yes	No	No	No
2	M7820	A2h	SLE78CLX1440PM	144	280	8	Yes	Yes	Yes	No	No
3	M7820	A3h	SLE78CLX1440PS	144	280	8	Yes	Yes	No	Yes	No
4	M7820	A4h	SLE78CLX1600P	160	280	8	Yes	Yes	No	No	No
5	M7820	A5h	SLE78CLX1600PM	160	280	8	Yes	Yes	Yes	No	No
6	M7820	A6h	SLE78CLX1600PS	160	280	8	Yes	Yes	No	Yes	No
7	M7820	A7h	SLE78CLX1280P	128	280	8	Yes	Yes	No	No	No
8	M7820	A8h	SLE78CLX1000P	100	280	8	Yes	Yes	No	No	No
9	M7820	A9h	SLE78CLX800P	80	280	8	Yes	Yes	No	No	No
10	M7820	AAh	SLE78CLX800PS	80	280	8	Yes	Yes	No	Yes	No
11	M7820	ABh	SLE78CLX800PM	80	280	8	Yes	Yes	Yes	No	No
12	M7820	ACH	SLE78CLX802P	80	216	8	Yes	Yes	No	No	No
13	M7820	ADh	SLE78CLX802PM	80	216	8	Yes	Yes	Yes	No	No
14	M7820	Aeh	SLE78CLX780P	78	280	8	Yes	Yes	No	No	No
15	M7820	Afh	SLE78CLX480P	48	280	8	Yes	Yes	No	No	No
16	M7820	B0h	SLE78CLX480PM	48	280	8	Yes	Yes	Yes	No	No
17	M7820	B1h	SLE78CLX360P	36	280	8	Yes	Yes	No	No	No
18	M7820	B2h	SLE78CLX360PM	36	280	8	Yes	Yes	Yes	No	No
19	M7820	B3h	SLE78CLX360PS	36	280	8	Yes	Yes	No	Yes	No
20	M7820	9Bh	SLE78CDX1440PSM	144	280	8	No/Yes(2)	No/Yes(2)	No/Yes(2)	No/Yes(2)	No/Yes(2)
21	M7820	9Ch	SLE97144SE	144	280	8	No	No	Yes	No	Yes
22	M7820	95h	SLE97080SE	80	280	8	No	No	Yes	No	Yes

No.	Product Code	Chip Identifier Byte	Sales Name	NVM(1) kBytes	ROM(1) kBytes	XRAM(1) kBytes	ISO 7816(2)	ISO 14443	Mifare compatible Interface	ISO/IEC18092 Passive mode	DCLB
23	M7820	96h	SLE97144SD	144	280	8	Yes	Yes	Yes	Yes	No
24	M7820	97h	SLE97080SD	80	280	8	Yes	Yes	Yes	Yes	No
25	M7820	98h	SLE 78CLFX1600P	160	0	8	Yes	Yes	No	No	No
26	M7820	99h	SLE 78CLFX1600PM	160	0	8	Yes	Yes	Yes	No	No
27	M7820	9Ah	SLE 78CLFX1600PSM	160	0	8	Yes	Yes	Yes	Yes	No

(1) Depicts the size of user available memory which is defined by blocking.

(2) Availability depends on procurement order: If the DCLB mode is chosen, contactless communication using the antenna and ISO7816 communication are out of operation. If the CL-interface is chosen, the DCLB mode communication is out of operation.

The above listed Chip Identifier Bytes show the TOE derivatives with the belonging configuration as defined today. Depending on the market demands new TOE derivatives with new Chip Identifier Bytes can be added over time and may be subject of additional certification processes (i.e. assurance continuity processes). This is for example to include additional interface options such as a Mifare compatible and/or the DCLB interface functionalities in future products. The blocking mechanism is also part of the evaluation. Each new chip configuration receives an own Chip Identifier Byte.

The TOE consists of the hardware part, the firmware parts and the software parts.

The software parts are differentiated into:

the cryptographic libraries RSA<sup>2</sup>, EC<sup>3</sup> and SHA-2<sup>4</sup> and the supporting libraries Toolbox and Base.

RSA, EC, SHA-2 and Toolbox provide certain functionality via an API to the Smartcard Embedded Software. The Base Library is only used internally by the RSA, EC and Toolbox libraries and has no user interface. If none the three libraries RSA, EC and Toolbox is delivered, also the Base Library is not on board. The SHA-2 library does not use the Base Library.

The firmware parts are the RMS library, the Service Algorithm Minimal (SAM), the STS firmware for test purpose (see chapter 2.2.2), providing some functionality via an API to the Smartcard Embedded Software, the Flash Loader for downloading user software to the NVM and the Mifare compatible software interface. The STS is implemented in a separated Test-ROM being part of the TOE.

The Smartcard Embedded Software, i.e. the operating system and applications are not part of the TOE.

The TOE can be delivered including - in free combinations - or not including any of the functionality of the cryptographic libraries EC, RSA, SHA-2 and the supporting Toolbox library. If RSA or EC or Toolbox is delivered, automatically the Base Library is part of the shipment too.

If the user decides not to use one or all of the crypto library(s), the specific library(s) is (are) not delivered to the user and the accompanying "Additional Specific Security Functionality (O.Add-Functions)" *Rivest-Shamir-Adleman (RSA)* and/ or *EC and/or SHA-2* is/are not provided by the TOE.

The Toolbox library provides the user optionally basic arithmetic and modular arithmetic operations, in order to support user software development using long integer operations. These basic arithmetic operations do not provide any security functionality, implement no security mechanism, and do not provide additional specific security functionality - as defined for the cryptographic libraries.

The user developed software using the Toolbox basic operations is not part of the TOE.

The Base Library provides the low level interface to the asymmetric cryptographic coprocessor and has no user available interface. The base library does not provide any security functionality, implements no security mechanism, and does not provide additional specific security functionality.

Deselecting one of the libraries does not include the code implementing functionality, which the user decided not to use. Not including the code of the deselected functionality has no impact of any other security policy of the TOE; it is exactly equivalent to the situation where the user decides just not to use the functionality.

The RSA, EC, SHA-2 and Toolbox libraries can be implemented together with the Smartcard Embedded Software in the User-ROM mask or respectively loaded into the SOLID FLASH™. This holds also for the Base Library, if the RSA, EC or Toolbox or combinations hereof is/are part of the shipment.

All other Smartcard Embedded Software does not belong to the TOE and is not subject of the evaluation.

## 1.2 Target of Evaluation overview

The TOE comprises the Infineon Technologies Dual Interface Security Controller M7820 with specific IC dedicated software and optional cryptographic RSA, EC, SHA-2 and Toolbox libraries.

---

2 Rivest-Shamir-Adleman asymmetric cryptographic algorithm

3 The Elliptic Curve Cryptography is abbreviated with EC only in the further, in order to avoid conflicts with the abbreviation for the Error Correction Code ECC.

4 SHA Secure Hash Algorithm

The TOE is a member of the Infineon Technologies AG high security controller family SLE70 meeting high requirements in terms of performance and security in context of the application. A summary product description is given in this Security Target (ST).

The SLE70 family provides a common architecture upon which specific products can be tailored for different markets.

The TOE is intended to be used in any applications and devices with highest security requirements. For example in smart cards and also in other applications, such as secure element in various devices. This product family features the Integrity Guard security philosophy focusing on data integrity. By that three main principles are combined in close synergy: full error detection, full encryption and intelligent active shielding.

With these capabilities this TOE can be used almost everywhere, where highly secure applications are in use and of course in any other application as well. This TOE is deemed for governmental, corporate, transport and payment markets, or wherever a secure root of trust is required. Various types of applications can use this TOE, for example in closed loop logical access controls, physical access controls, secure internet access control and internet authentication, or as multi-application token or simply as encrypted storage.

This dual interface controller is able to communicate using either the contact based or the contactless interface. The implemented dual interface provides a maximum flexibility in using following communication protocols respectively methods:

- ISO 7816
- ISO 14443 Type A
- ISO14443 Type B
- ISO/IEC 18092 passive mode
- Mifare compatible Interface or the
- Digital Contactless Bridge mode (DCLB)
- Advance Communication Mode (ACM).

The DCLB mode is provided by the specific TOE derivatives as listed in Table 5 and enables the use of an external analogue interface or near field communication (NFC) modem via the communication pads. Those external analogue modems are typically deemed for applications running in mobile devices and are not part of this TOE. In case of the available DCLB mode, the part of the contactless interface using the external antenna is out of operation.

The availability of the DCLB mode is configured during TOE production and depends on the customer order. Regarding the DCLB enabled derivatives it depends on the operating system of how the pads are used.

Independently from the used contactless protocol, the RFI provides also the option for Buffered Data Transfer BDT or Direct Data Transfer DDT. Using the BDT mode the CPU is in sleep or halt mode and in DDT the CPU is active during data transfer. The difference between the modes is the power consumption and the time required. A further communication option is the Advanced Communication Mode (ACM) allowing for very high bit rates according to ISO/IEC 14443-2AM3: Information Technology – Identification Cards – Contactless Integrated Circuit(s) Cards - Proximity Cards; Part 2: Radio frequency power and signal interface [2010]; AMENDMENT 3: Bits rates of  $fc/8$ ,  $fc/4$  and  $fc/2$ .

The TOE provides a real 16-bit CPU-architecture. The major components of the core system are the two CPUs (Central Processing Units), the MMU (Memory Management Unit) and MED (Memory Encryption/Decryption Unit). The two CPUs control each other in order to detect faults and serve by this for data integrity. The TOE implements a full 16 MByte linear addressable memory space for each privilege level, a simple scalable Memory Management concept and a scalable stack size. The flexible memory concept consists of ROM- and Flash-memory as part of the non-volatile memory (NVM), respectively SOLID FLASH™ NVM. For the SOLID FLASH™ NVM memory the Unified Channel Programming (UCP) memory technology is used.

The RMS library providing some functionality via an API to the Smartcard Embedded Software contains for example SOLID FLASH™ NVM service routines. The Service Algorithm provides functionality for the tearing save write into the SOLID FLASH™ NVM. The STS firmware is used for test purposes during start-up and the Flash Loader allows downloading user software to the NVM during the manufacturing process. The STS is implemented in a separated Test-ROM being part of the TOE.

The BSI has changed names and abbreviations for Random Number Generators, which is clarified as follows: The Physical True Random Number Generator (PTRNG), also named True Random Number Generator (TRNG) is a physical random number generator and meets the requirements of the functionality class AIS31 PTG.2, see [6]. It is used for provision of random number generation as a security service to the user and for internal purposes. The produced genuine random numbers can be used directly or as seed for the Deterministic Random Number Generator (DRNG), former named as Pseudo Random Number Generator (PRNG). The DRNG respectively PRNG is not in the scope of the evaluation. The TRNG respectively PTRNG is specially designed for smart cards, but can also be used in any other application where excellent physical random data are required.

The two cryptographic co-processors serve the need of modern cryptography: The symmetric co-processor (SCP) combines both AES and Triple-DES with dual-key or triple-key hardware acceleration. The Asymmetric Crypto Co-processor, called Crypto2304T in the following, is an optimized version with performance improvements for RSA and Elliptic Curve (EC) cryptography.

The optional software part of the TOE consists of the cryptographic RSA-, EC- and the SHA-2 libraries and the supporting Toolbox and Base libraries. If RSA or EC or Toolbox or combinations hereof are part of the shipment, automatically the Base Library is included.

The RSA library is used to provide a high level interface to RSA (Rivest, Shamir, Adleman) cryptography implemented on the hardware component Crypto2304T and includes countermeasures against SPA, DPA and DFA attacks. The routines are used for the generation of RSA Key Pairs (RsaKeyGen), the RSA signature verification (RsaVerify), the RSA signature generation (RsaSign) and the RSA modulus recalculation (RsaModulus). The hardware Crypto2304T unit provides the basic long number calculations (add, subtract, multiply, square with 1100 bit numbers) with high performance. The RSA library is delivered as object code and in this way integrated in the user software. The RSA library can perform RSA operations from 512 to 4096 bits. Following the BSI<sup>5</sup> recommendations, key lengths below 1024 bit are not included in the certificate.

The EC library is used to provide a high level interface to Elliptic Curve cryptography implemented on the hardware component Crypto2304T and includes countermeasures against SPA, DPA and DFA attacks. The routines are used for ECDSA signature generation, ECDSA signature verification, ECDSA key generation and Elliptic Curve Diffie-Hellman key agreement. In addition, the EC library provides an additional function for calculating primitive elliptic curve operations like EC Add and EC Double. EC curves over prime field  $F_p$ , as well as over  $GF(2^n)$  finite field are supported too.

The EC library is delivered as object code and in this way integrated in the user software. The certification covers the standard NIST [14] and Brainpool [15] Elliptic Curves with key lengths of 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 Bits, due to national AIS32 regulations by the BSI. Note that there are numerous other curve types, being also secure in terms of side channel attacks on this TOE, which can the user optionally add in the composition certification process.

---

<sup>5</sup> BSI Bundesamt für Sicherheit in der Informationstechnik – Federal office for information security



The SHA-library provides the calculation of a hash value of freely chosen data input in the CPU. The SHA-library is delivered as object code and is in this way available for the user software. This secure hash-algorithm SHA-2 is intended to be used for signature generation, verification and generic data integrity checks. The use for keyed hash operations like HMAC or similar security critical operations involving keys, is not subject of this TOE and requires specific security improvements and DPA analysis including the operating system, which is not part of this TOE.

The toolbox library does not provide cryptographic support or additional security functionality as it provides only the following basic long integer arithmetic and modular functions in software, supported by the cryptographic coprocessor: Addition, subtraction, division, multiplication, comparison, reduction, modular addition, modular subtraction, modular multiplication, modular inversion and modular exponentiation. No security relevant policy, mechanism or function is supported. The toolbox library is deemed for software developers as support for simplified implementation of long integer and modular arithmetic operations.

The Base Library provides the low level interface to the asymmetric cryptographic coprocessor and has no user available interface. The base library does not provide any security functionality, implements no security mechanism, and does not provide additional specific security functionality.

Note that this TOE can come with both cryptographic co-processors accessible, or with a blocked SCP or with a blocked Crypto2304T, or with both cryptographic co-processors blocked. The blocking depends on the user's choice. No accessibility of the deselected cryptographic co-processors is without impact on any other security policy of the TOE; it is exactly equivalent to the situation where the user decides just not to use the cryptographic co-processors. The TOE can be delivered without a specific library. In this case the TOE does not provide the Additional Specific Security Functionality Rivest-Shamir-Adleman Cryptography (RSA) or/and Elliptic Curve Cryptography (EC) or/and SHA-2.

To fulfill the highest security standards for smartcards today and also in the future, this TOE represents an entirely new security concept. This TOE utilizes digital security features to include customer friendly security, combined with a robust design overcoming the disadvantages on analogue protection technologies. The TOE provides full on-chip encryption covering the complete core, busses, memories and cryptographic co-processors leaving no plaintext on the chip. Therefore the attractiveness for attackers is extremely reduced as encrypted signals are of no use for the attacker – neither for manipulation nor for eavesdropping.

In addition, the TOE is equipped with a full error detection capability for the complete data path. The dual CPU approach allows error detection even while processing. A comparator detects whether a calculation was performed without errors. This approach does not leave any parts of the circuitry unprotected. The concept allows that the relevant attack scenarios are detected, whereas other conditions that would not lead to an error would mainly be ignored. And more, the TOE is equipped with signal protection implemented by an Infineon-specific shielding combined with secure wiring of security critical signals. Subsequently, an intelligent intrinsic shielding finishes the upper layers, finally providing the so called "I<sup>2</sup>-shield".

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

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

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



## 2 Target of Evaluation Description

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

### 2.1 TOE Definition

This TOE consists of smart card ICs (Security Dual Interface Controllers) meeting the highest requirements in terms of performance and security. They are manufactured by the Infineon Technologies AG in 120nm CMOS-technology (C120FL) with regard to the A11 design step. This TOE is intended to be used in smart cards or any other form factor for particularly security-relevant applications and for its previous use as developing platform for smart card operating systems or similar according to the lifecycle model from [1].

The term Smartcard Embedded Software is used in the following for all operating systems and applications stored and executed on the TOE regardless whether it is a smartcard or another application of form factor. The TOE is the platform for the Smartcard Embedded Software. The Smartcard Embedded Software itself is not part of the TOE.

The TOE consists of a core system, memories, co-processors, peripherals, security modules and analogue peripherals. The major components of the core system are the two CPUs (Central Processing Units), the MMU (Memory Management Unit) and MED (Memory Encryption/Decryption Unit). The co-processor block contains the processors for RSA/EC and DES/AES processing, while the peripheral block contains the random number generation and the external interfaces service. The peripheral block contains also the timers and a watchdog. All data of the memory block is encrypted and all memory types are equipped with an error detection code (EDC), the SOLID FLASH™ NVM in addition with an error correction code (ECC). The security modules serve for operation within the specified range and manage the alarms.

Note that there is a small set of sensors left in order to detect excessive deviations from the specified operational range, while not being over-sensitive. These digital features do not need adjustment or calibration and makes the chip even more robust. Conditions that would not be harmful for the operation would in most cases not influence the proper function.

This dual interface controller is able to communicate using either the contact based or the contactless interface. The implemented dual interface provides a maximum flexibility in using following communication protocols respectively methods:

- ISO 7816,
- ISO 14443 Type A
- ISO14443 Type B,
- ISO/IEC 18092 passive mode,
- Mifare compatible Interface or the
- Digital Contactless Bridge mode (DCLB)
- Advance Communication Mode (ACM).

The DCLB mode is provided by the specific TOE derivatives as listed in Table 5 and enables the use of an external analogue interface or near field communication (NFC) modem via the ISO-pads. Those external analogue modems are typically deemed for applications running in mobile devices and are not part of this TOE. In case of the available DCLB mode, the part of the contactless interface using the external antenna is out of operation. Whether the DCLB option is available or not is a configuration applied in TOE production which cannot be changed afterwards.

The availability of the DCLB mode is configured during TOE production and depends on the customer order. Regarding the DCLB enabled derivatives it depends on the operating system of how the pads are used.

Using the DCLB mode the external device feeds the analogue output of its modem – using the timing and protocol definitions – of ISO 14443 Type A or Type B, or ISO/IEC 18092 passive mode) via the ISO-pads to the digital part of the RF interface. It is also possible to bypass the coding/decoding and leave its interpretation up to the

software. By that further protocols can be implemented by the user software. Note that the feeding external analogue modem is not part of this TOE.

The availability of the DCLB mode is configured during TOE production and depends on the customer order. Regarding the DCLB enabled derivatives it depends on the operating system of how the pads are used.

Independently from the used contactless protocol, the RFI provides also the option for Buffered Data Transfer BDT or Direct Data Transfer DDT. Using the BDT mode the CPU is in sleep or halt mode and in DDT the CPU is active during data transfer. The difference between the modes is the power consumption and the time required. A further communication option is the Advanced Communication Mode (ACM) allowing for very high bit rates. More details are given in the confidential Errata Sheet as referenced in Table 1: Identification.

Possible interface options are contact-based (ISO7816) only, contactless only, contact-based (ISO7816) in parallel to contactless communication and DCLB communication. The DCLB communication excludes ISO7816 and contactless communication.

Supporting a Mifare compatible Interface application requires a dedicated small space of memory. Depending on user's choice, various Mifare compatible Interface memory sections of 1 up to 4 kByte each can be defined. The number and location of Mifare compatible Interface memory sections is simply limited by the available EEPROM space. The Mifare compatible Interface memory sections are read/write protected and are defined and generated by the user.

The CPU – here the two processors (CPU1 and CPU2) are seen from functional perspective as one - is compatible with the instruction set of the forerunner family 66-PE and is therefore also compatible to the SAB 80251 instruction set (8051 is a subset hereof) and is multiple times faster than the standard processor. It provides additional powerful instructions for smart card applications. It thus meets the requirements for the new generation of operating systems. Despite its compatibility the CPU implementation is entirely proprietary and not standard.

The CPU – here the two processors (CPU1 and CPU2) are seen from functional perspective as one - accesses the memory via the integrated Memory Encryption and Decryption unit (MED). The access rights of the application to the memories can be controlled with the memory management unit (MMU). Errors in the memories are automatically detected (EDC) and in terms of the SOLID FLASH™ NVM 1-Bit-errors are also corrected (ECC). The two processors of the CPU control each other in order to detect faults and maintain by this the data integrity. A comparator detects whether a calculation was performed without errors and allows error detection even while processing. Therefore the TOE is equipped with a full error detection capability for the complete data path, which does not leave any parts of the circuitry unprotected.

The controllers of this TOE store both code and data in a linear 16-MByte memory space, allowing direct access without the need to swap memory segments in and out of memory using a memory management unit.

The error detection unit (EDU) automatically manages the error detection of the individual memories and detects incorrect transfer of data between the memories by means of error code comparison.

The CACHE memory – or simply, the CACHE – is a high-speed memory-buffer located between the CPU and the (external) main memories holding a copy of some of the memory contents to enable access to the copy, which is considerably faster than retrieving the information from the main memory. In addition to its fast access speed, the CACHE also consumes less power than the main memories. All CACHE systems own their usefulness to the principle of locality, meaning that programs are inclined to utilize a particular section of the address space for their processing over a short period of time. By including most or all of such a specific area in the CACHE, system performance can be dramatically enhanced. The implemented post failure detection identifies and manages errors if appeared during storage.

The TRNG respectively PTRNG is specially designed for smart cards, but can also be used in any other application where excellent physical random data are required. The TRNG respectively PTRNG fulfils the requirements from the functionality class PTG.2 of the AIS31 and produces genuine random numbers which then can be used directly or as seed for the Deterministic Random Number Generator (DRNG), former named as Pseudo Random Number Generator (PRNG). The DRNG respectively PRNG is not in the scope of the evaluation.

The implemented sleep mode logic (clock stop mode per ISO/IEC 7816-3) is used to reduce the overall power consumption. Contactless products provide a low-power halt mode for operation with reduced power.

The timer permits easy implementation of communication protocols such as T=1 and all other time-critical operations. The UART-controlled I/O interface allows the smart card controller and the terminal interface to be operated independently.

The Clock Unit (CLKU) supplies the clocks for all components of the TOE. The Clock Unit can work in an internal and external clock mode. When operating in the internal clock mode the system frequency may be varied in a

range of approximately 1 MHz up to 33 MHz in steps of roughly 1 MHz. This enables a programmer to choose the best-fitting frequency for an application in consideration of a potential current limit and a demanded application performance. The frequencies are derived from the 33 MHz clock of an internal VCO (VCOCLK), whereas the system clock (SYSCLK) may either be based on the internal 33 MHz VCO clock (VCOCLK) or on an external clock such as the clock of the CB interface (EXTCLK). In this external clock mode, the system clock is derived from an externally applied interface clock according to a defined dependency. The system frequency may be 1 up to 8 times the externally applied frequency but is of course limited to the maximum system frequency of 33 MHz.

Two co-processors for cryptographic operations are implemented on the TOE: The Crypto2304T for calculation of asymmetric algorithms like RSA and Elliptic Curve (EC) and the Symmetric Cryptographic Processor (SCP) for dual-key or triple-key triple-DES and AES calculations. These co-processors are especially designed for smart card applications with respect to the security and power consumption. The SCP module computes the complete DES algorithm within a few clock cycles and is especially designed to counter attacks like DPA, EMA and DFA.

Note that this TOE can come with both cryptographic co-processors accessible, or with a blocked symmetric cryptographic co-processor SCP or with a blocked asymmetric cryptographic co-processor Crypto2304T, or with both cryptographic co-processors blocked. The blocking depends on the customer demands prior to the production of the hardware. No accessibility of the deselected cryptographic co-processors is without impact on any other security policy of the TOE; it is exactly equivalent to the situation where the user decides just not to use the cryptographic co-processors.

The STS (self-test software), RMS (Resource Management System), Service Algorithm Minimal (SAM) and Flash Loader together compose the TOE firmware stored in the ROM and the patches hereof in the SOLID FLASH™ NVM. All mandatory functions for internal testing, production usage and start-up behavior (STS), and also the RMS and SAM functions are grouped together in a common privilege level. These privilege levels are protected by a hardwired Memory Management Unit (MMU) setting.

The user software can be implemented in various options depending on the user's choice and described in chapter 1.1. Thereby the user software can be implemented in the ROM and/or the NVM or coming without user software. In the latter case, the user downloads his entire software on his own using the Flash Loader software.

The TOE uses also Special Function Registers SFR. These SFR registers are used for general purposes and chip configuration. The start-up register values are stored in the SOLID FLASH™ NVM, in the configuration page area.

The bus system comprises two separate bus entities: a memory bus and a peripheral bus for high-speed communication with the peripherals.

An intelligent intrinsic shielding finishes the upper layers above security critical signals and wires, finally providing the so called "I<sup>2</sup>-shield".

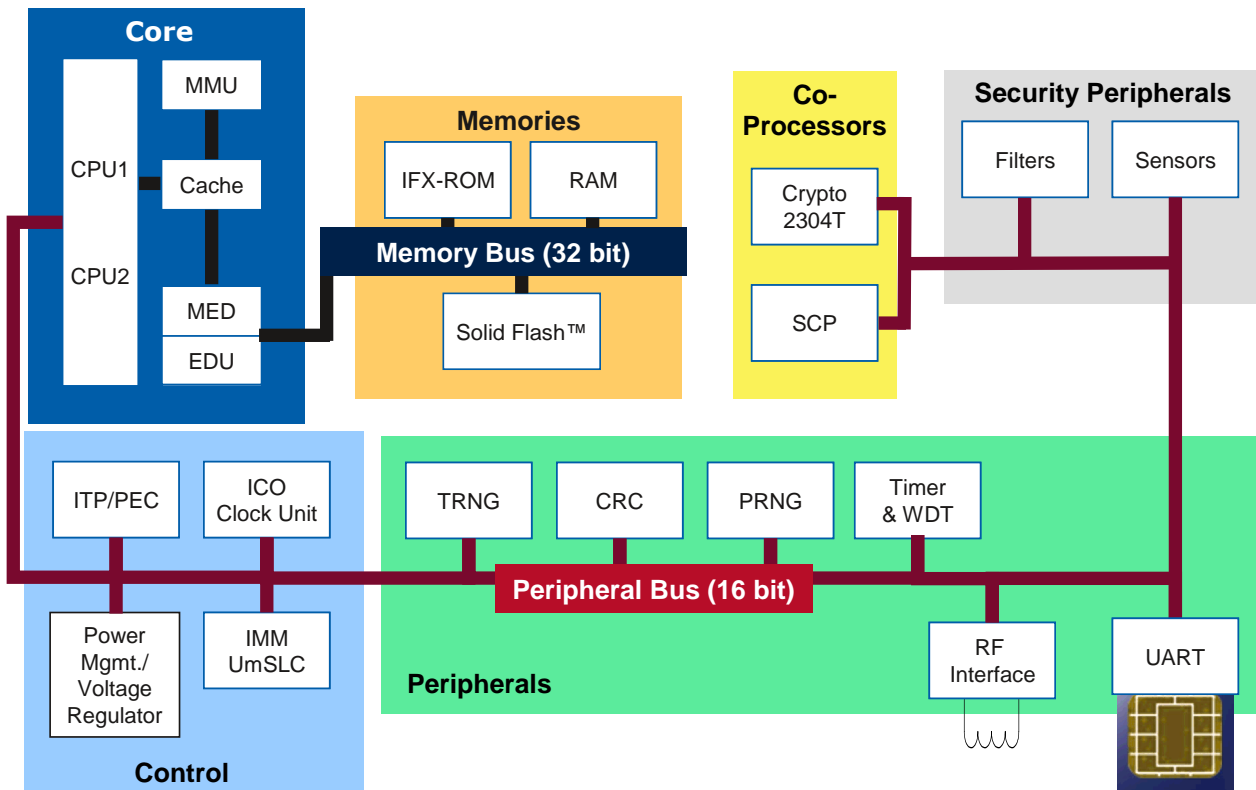
The following is a list of features provided by this TOE:

- 24-bit linear addressing
- Up to 16 MByte of addressable memory
- Register-based architecture (registers can be accessed as bytes, words (2 bytes), and doublewords (4 bytes))
- 2-stage instruction pipeline
- Extensive set of powerful instructions, including 16- and 32-bit arithmetic and logic instructions
- CACHE with single-cycle access searching
- 16-bit ALU
- Minimum instruction execution time of one clock

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

To sum up, the TOE is a powerful smart card dual interface IC with a large amount of memory and special peripheral devices with improved performance, optimized power consumption, free to choose contact based or contactless operation, at minimal chip size while implementing high security. It therefore constitutes the basis for future smart card applications.

Figure 1: Block diagram of the TOE



## 2.2 Scope of the TOE

The TOE comprises as one part the hardware of the smart card security controller in various configurations as listed in Table 4 and Table 4. All products of this TOE, including also the different configurations and resulting chip identifier bytes, are manufactured by Infineon Technologies AG. Note that future configurations of this TOE result in different chip identifier bytes which today are not listed in this ST. The listing of Table 5 contains therefore the product of this TOE as present today and covered by the certificate. New configuration can be added by additional certification processes, i.e. assurance continuity processes (maintenance). The various blocking options of memory sizes and coprocessors are done during the manufacturing process depending on the customer order and are subject of the evaluation. All resulting combinations of derivatives are subject of the certificate.

The second part of this TOE includes the parts of the associated firmware and software required for operation and cryptographic support. The documents as described in section 2.2.4 and listed in Table 1: Identification, are supplied as user guidance. In the following description, the term “manufacturer” is short for Infineon Technologies AG, the manufacturer of the TOE. The Smartcard Embedded Software respectively user software is not part of the TOE.

### 2.2.1 Hardware of the TOE

The hardware part of the TOE (see Figure 1) as defined in [1] is comprised of:

#### Core System

Proprietary CPU implementation of the Intel MCS251 standard architecture from functional perspective, represented by two CPUs from hardware perspective

CACHE with Post Failure Detection

Memory Encryption/Decryption Unit (MED) and Error Detection Unit (EDU)

Memory Management Unit (MMU)

#### Memories

Read-Only Memory (ROM)

Random Access Memory (RAM)

SOLID FLASH™ NVM memory equals in the meaning to the Electrical Erasable Programmable Read Only Memory (EEPROM)

Note that the TOE has implemented an SOLID FLASH™ NVM memory module, where parts of this memory module are configured to work as an EERPOM.

#### Peripherals

True Random Number Generator (TRNG) respectively Physical True Random Number Generator (PTRNG)

Deterministic Random Number Generator (DRNG) respectively Pseudo Random Number Generator (PRNG)

Watchdog and Timers

Universal Asynchronous Receiver/Transmitter (UART)

Checksum module (CRC)

RF interface (radio frequency power and signal interface)

#### Control

Dynamic Power Management

Internal Clock Oscillator (ICO)

Interrupt and Peripheral Event Channel Controller (ITP and PEC)

Interface Management Module (IMM)

User mode Security Life Control (UmSLC)

Voltage Regulator

#### Coprocessors

Crypto2304T for asymmetric algorithms like RSA and EC (optionally blocked)

Symmetric Crypto Co-processor for 3DES and AES Standards (optionally blocked)

#### Security Peripherals

Filters

Sensors

#### Buses

Memory Bus

Peripheral Bus

## 2.2.2 Firmware and software of the TOE

The entire firmware of the TOE consists of different parts:

One part comprises the RMS and SAM routines for SOLID FLASH™ NVM programming, security functions test, and random number online testing (Resource Management System, IC Dedicated Support Software in PP [1]).

The RMS and SAM routines are stored from Infineon Technologies AG in a reserved area of the normal user ROM.

The second part is the STS, consisting of test and initialization routines (Self-Test Software, IC Dedicated Test Software in PP [1]). The STS routines are stored in the especially protected test ROM and are not accessible for the user software.

The third part is the Flash Loader, a piece of software located in the user-ROM and allowing downloading the user software or parts of it to the SOLID FLASH™ NVM memory. After completion of the download the Flash Loader can be permanently deactivated by the user.

The fourth part is the Mifare compatible interface routines called via RMS routines if the Mifare compatible interface option is active. Note that the Mifare compatible Interface portion is always present but deactivated in case of the non-Mifare compatible Interface derivatives. Thus the user interface is identically in both cases and subsequently the Mifare compatible Interface routines can be called in each of the derivatives. In case Mifare compatible Interface routines are called in derivatives without Mifare compatible Interface a dedicated error code is returned and in case of the Mifare compatible Interface derivate the according function is performed.

All parts of the firmware above are combined together by the TOE generation process to a single file and stored then in the data files, the TOE is produced from. This comprises the firmware files for the ROM, where only Infineon Technologies AG has access, as well as the data to be flashed in the SOLID FLASH™ NVM.

The optional software part of the TOE consists of the cryptographic libraries the RSA-, the EC, the SHA-2 library and the supporting Toolbox and Base libraries.

The RSA library is used to provide a high level interface to the RSA cryptography implemented on the hardware component Crypto2304T and includes countermeasures against SPA, DPA and DFA attacks. The routines are used for the generation of RSA Key Pairs (RsaKeyGen), the RSA signature verification (RsaVerify), the RSA signature generation (RsaSign) and the RSA modulus recalculation (RsaModulus). The module provides the basic long number calculations (add, subtract, multiply, square with 1100 bit numbers) with high performance.

The RSA library is delivered as object code and in this way integrated in the user software. The RSA library can perform RSA operations from 512 to 4096 bits. Depending on the customer's choice, the TOE can be delivered with the 4096 code portion or with the 2048 code portion only. The 2048 code portion is included in both. Part of the evaluation are the RSA straight operations with key length from 1024 bits to 2048 bits, and the RSA CRT<sup>6</sup> operations with key lengths of 1024 Bits to 4096 Bits.

The EC library is used to provide a high level interface to Elliptic Curve cryptography and includes countermeasures against SPA, DPA and DFA attacks. The routines are used for ECDSA signature generation, ECDSA signature verification, ECDSA key generation and Elliptic Curve Diffie-Hellman key agreement. In addition, the EC library provides an interface to an addition function for primitive elliptic curve operations like ECC Add and ECC Double. EC curves over prime field  $F_p$ , as well as over  $GF(2^n)$  finite field are supported too. Note that the according user guidance abbreviates the Elliptic Curve cryptographic functions with ECC.

The EC library is delivered as object code and in this way integrated in the user software. The certification covers the standard NIST [14] and Brainpool [15] Elliptic Curves with key lengths of 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 Bits, due to national AIS32 regulations by the BSI. Note that numerous other curve types being also secure in terms of side channel attacks on this TOE exists, which can the user optionally add in the composition certification process.

The SHA-2 library provides the calculation of a hash value of freely chosen data input in the CPU. The SHA-2 library is delivered as object code and is in this way available for the user software. This secure hash-algorithm SHA-2 is intended to be used for signature generation, verification and generic data integrity checks. The use for keyed hash operations like HMAC or similar security critical operations involving keys, is not subject of this TOE and requires specific security improvements and DPA analysis including the operating system, which is not part of this TOE.

The toolbox library does not provide cryptographic support or additional security functionality as it provides only the following basic long integer arithmetic and modular functions in software, supported by the

---

<sup>6</sup> CRT: Chinese Remainder Theorem



cryptographic coprocessor: Addition, subtraction, division, multiplication, comparison, reduction, modular addition, modular subtraction, modular multiplication, modular inversion and modular exponentiation. No security relevant policy, mechanism or function is supported. The toolbox library is deemed for software developers as support for simplified implementation of long integer and modular arithmetic operations.

The Base Library provides the low level interface to the asymmetric cryptographic coprocessor and has no user available interface. The base library does not provide any security functionality, implements no security mechanism, and does not provide additional specific security functionality.

Note 1:

The cryptographic libraries RSA, EC, SHA-2 and the Toolbox libraries are delivery options. Therefore the TOE may come with free combinations of or without these libraries. In the case of coming without one or any combination of these libraries the TOE does not provide the Additional Specific Security Functionality Rivest-Shamir-Adleman Cryptography (RSA) and/or Elliptic Curve Cryptography (EC) and/or SHA-2.

End of note.

### 2.2.3 Interfaces of the TOE

- The physical interface of the TOE to the external environment is the entire surface of the IC.
- The electrical interface of the TOE to the external environment is constituted by the pads of the chip, particularly the contacted RES, I/O, CLK lines and supply lines VCC and GND, as well as by the contactless RF interface. The contact based communication is according to ISO 7816/ETSI/EMV if no DCLB mode is chosen.  
For the case the DCLB mode is active, the pads are also used to communicate with an external analogue modem. The radio frequency interface protocols are then driven by the external modem via the IO pad: During reception the signal fed to the IO is directly forwarded to the decoding unit. During transmission, the digital modulation signal generated by the IC is fed to the IO. The part of the RF interface normally using the antenna is then out of operation.
- The RF interface (radio frequency power and signal interface) enables contactless communication between a PICC (proximity integration chip card, PICC) and a PCD reader/writer (proximity coupling device, PCD), if no DCLB mode was chosen at TOE start-up.  
Power supply is received and data are received or transmitted by an antenna which consists of a coil with a few turns directly connected to the IC. Depending on customer orders the contactless interface options are set by means of blocking and delivered as depicted in Table 5: Today's predefined hardware configurations and derivatives.
- The data-oriented I/O interface to the TOE is formed by the I/O pad and by the various RF options.
- The interface to the firmware is constituted by special registers used for hardware configuration and control (Special Function Registers, SFR).
- The interface of the TOE to the operating system is constituted on one hand by the RMS routine calls and on the other by the instruction set of the TOE.
- The interface of the TOE to the test routines is formed by the STS test routine call, i.e. entry to test mode (STS-TM entry).
- The interface to the RSA calculations is defined from the RSA library interface.
- The interface to the EC calculations is defined from the EC library interface
- The interface to the SHA-2 calculation is defined from the SHA-2 library interface.
- The interface to the Toolbox basic arithmetic functions is defined from the Toolbox library.

Note that the interfaces to the cryptographic libraries (RSA, EC and SHA-2) and toolbox library are optionally, as it depends on the user order.

### 2.2.4 Guidance documentation

The guidance documentation consists of the:

- SLx 70 Family Hardware Reference Manual
- SLx 70 Family Production and Personalization User's Manual

- SLE 70 Family Programmer's Reference User's Manual
- M7820 Controller Family for Security Application, Security Guidelines
- M7801/M7820 Controller Family for Security Applications Errata Sheet, containing the description of all interfaces of the software to the hardware relevant for programming the TOE. The SLE70 Family Errata Sheet can be changed during the life cycle of the TOE. This is reported in a monthly updated list [5] provided from Infineon Technologies AG to the user.
- Crypto@2304T User Manual, describing the architecture of cryptographic coprocessor on register level. It also provides a functional description of the register architecture, instruction set and gives programming guidance.
- SLE 70 Asymmetric Crypto Library for Crypto2304T, RSA / ECC / Toolbox, User Interface, contains all interfaces of the cryptographic RSA- and EC libraries, as well as of the Toolbox library. This document is only delivered to the user in case the RSA library and/or the EC library and/or the Toolbox library is/are part of the delivered TOE. This document has been updated with an additional errata sheet section.
- SLx 70 Family, Secure Hash Algorithm SHA-2, (SHA 256/224, SHA 512/384), Confidential Application Note, Library Version 1.01, contains all interfaces of the SHA-2 library and is only delivered to the user in case the SHA-2 library is part of the delivered TOE.

The exact versions are referenced in Table 1: Identification.

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



### 2.2.5 Forms of delivery

The TOE can be delivered in any form of complete module, package, with or without inlay mounting, in form of plain wafers or in an IC case (e.g. DSO20) or in bare dies. The delivery can therefore be at the end of phase 3 or at the end of phase 4 which can also include pre-personalization steps according to PP [1]. In this document are always both cases mentioned to avoid incorrectness but from the security policy point of view the two cases are identical.

From a security policy point of view the different forms of delivery do not have any impact.

The delivery to the software developer (phase 2 → phase 1) contains the development package, which is delivered in electronic form. It contains the documents as described above, the development and debugging tools.

Part of the software delivery could also be the Flash Loader program, provided by Infineon Technologies, running on the TOE and receiving via the UART interface the transmitted information of the user software to be loaded into the SOLID FLASH™ NVM memory. The download is only possible after successful authentication. The user software can also be downloaded in an encrypted way. In addition, the user can permanently block further use of the Flash Loader. Whether the Flash Loader program is present or not depends on the procurement order.

The table as follows provides an overview about form and method of TOE deliveries:

Table 6: TOE forms and delivery

TOE Component	Delivered Format	Delivery Method	Comment
<b>Hardware</b>			
M7820 A11	Wafer, bare dies, modules, IC cases, with or without antenna in various packages	Postal transfer in cages	All materials are delivered to distribution centers in locked cages.
<b>Firmware</b>			
All	–	–	Stored on the delivered hardware.
<b>Software</b>			
All software libraries	Precompiled object format code – L251 file type with ending “.lib”	Secured download as zipped-file (1)	In addition, the user receives the source code as well.
<b>Guidance Documentation</b>			
All User Guidance documents	Personalized pdf	Secured download (1) or on demand by encrypted email.	–

(1) Authorized users having an account on the secure server can login and download the according document- respectively software files

## 2.2.6 Production sites

The TOE is produced in the IC production site Dresden as listed below. Other involved production sites during the life cycle handle subsequent production steps only. To distinguish the different production sites of various products in the field, the site is coded into the Chip Ident Mode data. The exact coding of the chip identification data is described in [7].

The delivery measures are described in the ALC\_DVS aspect.

Table 7: Production site in chip identification

Production Site	Chip Identification
Dresden, Germany	Bits 7:4 of batch byte number 06: 0010

### 3 Conformance Claims (ASE\_CCL)

#### 3.1 CC Conformance Claim

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

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

#### 3.2 PP Claim

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

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

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

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

Table 8: Augmentations of the assurance level of the TOE

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

#### 3.3 Package Claim

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

The assurance level for the TOE is EAL5 augmented with the components ALC\_DVS.2 and AVA\_VAN.5.

### 3.4 Conformance Rationale

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

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

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

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

Security Problem Definition:

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

Conformance Rationale:

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

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

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

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

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

All security functional requirements defined in the PP [1] are included and completely defined in this ST. The security functional requirements listed in the following are all taken from Common Criteria part 2 [3] and additionally included and completely defined in this ST:

- FDP\_ACC.1 "Subset access control"
- FDP\_ACF.1 "Security attribute based access control"
- FMT\_MSA.1 "Management of security attributes"
- FMT\_MSA.3 "Static attribute initialization"
- FMT\_SMF.1 "Specification of Management functions"
- FCS\_COP.1 "Cryptographic support"
- FCS\_CKM.1 "Cryptographic key generation"

- FDP\_SDI.1 “Stored data integrity monitoring
- FDP\_SDI.2 “Stored data integrity monitoring and action

The security functional requirement

- FPT\_TST.2 “Subset TOE security testing“(Requirement from [3])
- FCS\_RNG.1 “Generation of Random Numbers”

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

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

The Assurance Requirements of the TOE obtain the Evaluation Assurance Level 5 augmented with the assurance components ALC\_DVS.2 and AVA\_VAN.5 for the TOE.

### 3.5 Application Notes

The functional requirement FCS\_RNG.1 is a refinement of the FCS\_RNG.1 defined in the Protection Profile [1] according to “Anwendungshinweise und Interpretationen zum Schema (AIS)” respectively “Functionality classes and evaluation methodology for physical random number generators”, AIS31 [6].

## 4 Security Problem Definition (ASE\_SPD)

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

### 4.1 Threats

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

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

Table 9: Threats according PP [1]

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

#### 4.1.1 Additional Threat due to TOE specific Functionality

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

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

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

T.Mem-Access                      Memory Access Violation

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

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

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

For details see PP [1] section 3.2.

### 4.1.2 Assets regarding the Threats

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

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

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

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

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

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

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

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

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

For details see PP [1] section 3.1.

## 4.2 Organizational Security Policies

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

P.Process-TOE Protection during TOE Development and Production

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

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

Table 11: Organizational Security Policies according PP [1]

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

### 4.2.1 Augmented Organizational Security Policy

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

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

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

P.Add-Functions	<p>Additional Specific Security Functionality</p> <p>The TOE shall provide the following specific security functionality to the Smartcard Embedded Software:</p> <ul style="list-style-type: none"> <li>• Advanced Encryption Standard (AES)</li> <li>• Triple Data Encryption Standard (3DES)</li> <li>• Rivest-Shamir-Adleman Cryptography (RSA),</li> <li>• Elliptic Curve Cryptography (EC)</li> <li>• Secure Hash Algorithm SHA-2</li> </ul>
-----------------	---

Note 2:

The cryptographic libraries RSA, EC, SHA-2 and the Toolbox library are delivery options. Therefore the TOE may come with free combinations of or even without these libraries. In the case of coming without one or any combination of the cryptographic libraries RSA, EC and SHA-2, the TOE does not provide the Additional Specific Security Functionality Rivest-Shamir-Adleman Cryptography (RSA) and/or Elliptic Curve Cryptography (EC) and/or SHA-2. The Toolbox library is no cryptographic library and provides no additional specific security functionality. If RSA, EC or Toolbox are part of the shipment, the Base Library is automatically included. The Base Library does not provide additional specific functionality.

End of note.

Note 3:

This TOE can come with both crypto co-processors accessible, or with a blocked SCP or with a blocked Crypto2304T, or with both crypto co-processors blocked. The blocking depends on the customer demands prior to the production of the hardware. In case the SCP is blocked, no AES and 3DES computation supported by hardware is possible. In case the Crypto2304T is blocked, no RSA and EC computation supported by hardware is possible. The use of the SHA-2 library is also possible with both crypto coprocessors blocked. No accessibility of the deselected cryptographic co-processors is without impact on any other security policy of the TOE; it is exactly equivalent to the situation where the user decides just not to use the cryptographic co-processors.

End of note.

### 4.3 Assumptions

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

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

A.Process-Sec-IC	<p>Protection during Packaging, Finishing and Personalization</p> <p>It is assumed that security procedures are used after delivery of the TOE by the TOE Manufacturer up to delivery to the end-consumer to maintain confidentiality and integrity of the TOE and of its manufacturing and test data (to prevent any possible copy, modification, retention, theft or unauthorised use).</p>
A.Plat-Appl	<p>Usage of Hardware Platform</p> <p>The Security IC Embedded Software is designed so that the requirements from the following documents are met: (i) TOE guidance documents (refer to the Common Criteria assurance class AGD) such as the hardware data sheet, and</p>



the hardware application notes, and (ii) findings of the TOE evaluation reports relevant for the Security IC Embedded Software as documented in the certification report.

A.Resp-Appl Treatment of User Data

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

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

Table 12: Assumption according PP [1]

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

### 4.3.1 Augmented Assumptions

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

A.Key-Function

Usage of Key-dependent Functions

Key-dependent functions (if any) shall be implemented in the Smartcard Embedded Software in a way that they are not susceptible to leakage attacks (as described under T.Leak-Inherent and T.Leak-Forced).

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

For details see PP [1] section 3.4.

## 5 Security objectives (ASE\_OBJ)

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

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

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

### 5.1 Security objectives for the TOE

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

Table 13: Objectives for the TOE according to PP [1]

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

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

O.Add-Functions

Additional Specific Security Functionality

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

- Advanced Encryption Standard (AES)
- Triple Data Encryption Standard (3DES),
- Rivest-Shamir-Adleman (RSA)
- Elliptic Curve Cryptography (EC)
- Secure Hash Algorithm (SHA-2)

Note 4:

The cryptographic libraries RSA, EC, SHA-2 and the Toolbox library are delivery options. If one of the libraries RSA, EC and Toolbox or combination hereof are delivered, the Base Lib is automatically part of it. Therefore the TOE may come with free combinations of or even without these libraries. In the case of coming without one or any combination of the cryptographic libraries RSA, EC and SHA-2, the TOE does not provide the Additional Specific Security Functionality Rivest-Shamir-Adleman Cryptography (RSA) and/or Elliptic Curve Cryptography (EC) and/or SHA-2. The Toolbox and Base Library are no cryptographic libraries and provide no additional specific security functionality.

End of note.

Note 5:

This TOE can come with both crypto co-processors accessible, or with a blocked SCP or with a blocked Crypto2304T, or with both crypto co-processors blocked. The blocking depends on the customer demands prior to the production of the hardware. In case the SCP is blocked, no AES and 3DES computation supported by hardware is possible. In case the Crypto2304T is blocked, no RSA and EC computation supported by hardware is possible. The use of the SHA-2 library is also possible with both crypto coprocessors blocked. No accessibility of the deselected cryptographic co-processors is without impact on any other security policy of the TOE; it is exactly equivalent to the situation where the user decides just not to use the cryptographic co-processors.  
End of note.

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

O.Mem-Access

Area based Memory Access Control

The TOE must provide the Smartcard Embedded Software with the capability to define restricted access memory areas. The TOE must then enforce the partitioning of such memory areas so that access of software to memory areas and privilege levels is controlled as required, for example, in a multi-application environment.

Table 14: Additional objectives due to TOE specific functions and augmentations

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

## 5.2 Security Objectives for the development and operational Environment

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

Table 15: Security objectives for the environment according to PP [1]

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

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

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

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

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

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

function of the TOE and use keys and functions appropriately in order to ensure the strength of cryptographic operation.

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

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

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

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

## 5.3 Security Objectives Rationale

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

Table 16: Security Objective Rationale

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

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

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

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

Compared to the PP [1] a clarification has been made for the security objective “Treatment of User Data (OE.Resp-Appl)”: By definition cipher or plain text data and cryptographic keys are User Data. So, the Smartcard Embedded Software will protect such data if required and use keys and functions appropriately in order to ensure the strength of cryptographic operation. Quality and confidentiality must be maintained for keys that are imported and/or derived from other keys. This implies that appropriate key management has to be realised in the

environment. That is expressed by the assumption A.Key—Function which is covered from OE.Resp—Appl. These measures make sure that the assumption A.Resp—Appl is still covered by the security objective OE.Resp—Appl although additional functions are being supported according to P.Add-Functions.

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

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

## 6 Extended Component Definition (ASE\_ECD)

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

- the family **FCS\_RNG** at the class FCS Cryptographic Support
- the family **FMT\_LIM** at the class FMT Security Management
- the family **FAU\_SAS** at the class FAU Security Audit
- the component **FPT\_TST.2** at the class FPT Protection of the TSF

The extended components FMT\_LIM and FAU\_SAS are defined and described in PP [1] section 5. The components FCS\_RNG and FPT\_TST.2 are defined in the following.

### 6.1 Component “Subset TOE security testing (FPT\_TST)”

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

Part 2 of the Common Criteria provides the security functional component “TSF testing (FPT\_TST.1)”. The component FPT\_TST.1 provides the ability to test the TSF’s correct operation.

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

### 6.2 Definition of FPT\_TST.2

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

The functional component “Subset TOE testing (FPT\_TST.2)” is specified as follows (Common Criteria Part 2 extended).

### 6.3 TSF self-test (FPT\_TST)

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

Component leveling



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

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

Management: FPT\_TST.2

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

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

Audit: FPT\_TST.2

There are no auditable events foreseen.

**FPT\_TST.2** Subset TOE testing

Hierarchical to: No other components.

Dependencies: No dependencies.

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

### 6.4 Family “Generation of Random Numbers (FCS\_RNG)”

The family “Generation of Random Numbers (FCS\_RNG.1)” has to be newly created according the new version of the “Anwendungshinweise und Interpretationen zum Schema (AIS)” respectively “Functionality classes and evaluation methodology for physical random number generators”, AIS31, [6]. This security functional component is used instead of the functional component FCS\_RNG.1 defined in the protection profile [1].

The family “Generation of Random Numbers (FCS\_RNG.1)” is specified as follows (Common Criteria Part 2 extended).



## 6.5 Definition of FCS\_RNG.1

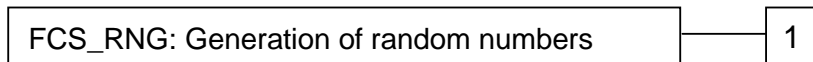
This section describes the functional requirements for the generation of random numbers, which may be used as secrets for cryptographic purposes or authentication. The IT security functional requirements for the TOE are defined in an additional family (FCS\_RNG) of the Class FCS (Cryptographic support).

### FCS\_RNG Generation of random numbers

Family Behaviour

This family defines quality requirements for the generation of random numbers that are intended to be used for cryptographic purposes.

Component leveling:



FCS\_RNG.1: Generation of random numbers, requires that the random number generator implements defined security capabilities and that the random numbers meet a defined quality metric.

Management: FCS\_RNG.1

There are no management activities foreseen.

Audit: FCS\_RNG.1

There are no actions defined to be auditable.

<b>FCS_RNG.1</b>	Random number generation
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FCS_RNG.1.1:	The TSF shall provide a [selection: <i>physical, non-physical true, deterministic, hybrid physical, hybrid deterministic</i> ] random number generator that implements: [assignment: <i>list of security capabilities</i> ].
FCS_RNG.1.2:	The TSF shall provide random numbers that meet [assignment: <i>a defined quality metric</i> ].
Application Note 1:	The functional requirement FCS_RNG.1 is a refinement of the FCS_RNG.1 defined in the Protection Profile [1] according to "Anwendungshinweise und Interpretationen zum Schema (AIS)" respectively "Functionality classes and evaluation methodology for physical random number generators", AIS31 [6].

## 7 Security Requirements (ASE\_REQ)

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

### 7.1 TOE Security Functional Requirements

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

The Table 15 provides an overview of the functional security requirements of the TOE, defined in the in PP [1] section 6.1. In the last column it is marked if the requirement is refined. The refinements are also valid for this ST.

Table 17: Security functional requirements defined in PP [1]

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

The Table 16 provides an overview about the augmented security functional requirements, which are added additional to the TOE and defined in this ST All requirements are taken from Common Criteria Part 2 [3], with the exception of the requirement FPT\_TST.2 and FCS\_RNG, which are defined in this ST completely.

Table 18: Augmented security functional requirements

Security Functional Requirement	
FPT_TST.2	“Subset TOE security testing“
FDP_ACC.1	“Subset access control“
FDP_ACF.1	“Security attribute based access control“
FMT_MSA.1	“Management of security attributes“
FMT_MSA.3	“Static attribute initialization“
FMT_SMF.1	“Specification of Management functions“
FCS_COP.1	“Cryptographic support“
FCS_CKM.1	“Cryptographic key management“
FDP_SDI.1	“Stored data integrity monitoring“
FDP_SDI.2	“Stored data integrity monitoring and action“
FCS_RNG.1	“Quality metric for random numbers“

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

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

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

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

## 7.1.1 Extended Components FCS\_RNG.1 and FAU\_SAS.1

### 7.1.1.1 FCS\_RNG

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

<b>FCS_RNG.1</b>	Random Number Generation
Hierarchical to:	No other components
Dependencies:	No dependencies
FCS_RNG.1	Random numbers generation Class PTG.2 according to [6]
FCS_RNG.1.1	The TSF shall provide a <i>physical</i> random number generator that implements:
PTG.2.1	<i>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.</i>
PTG.2.2	<i>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.</i>
PTG.2.3	<i>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.</i>
PTG.2.4	<i>The online test procedure shall be effective to detect non-tolerable weaknesses of the random numbers soon.</i>
PTG.2.5	<i>The online test procedure checks the quality of the raw random number sequence. It is triggered 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.</i>
FCS_RNG.1.2	The TSF shall provide <i>numbers in the format 8- or 16-bit</i> that meet
PTG.2.6	<i>Test procedure A, as defined in [6] does not distinguish the internal random numbers from output sequences of an ideal RNG.</i>
PTG.2.7	<i>The average Shannon entropy per internal random bit exceeds 0.997.</i>
Application Note 2:	The functional requirement FCS_RNG.1 is a refinement of the FCS_RNG.1 defined in the Protection Profile [1] according to "Anwendungshinweise und Interpretationen zum Schema (AIS)" respectively "A proposal for: Functionality classes for random number generators" [6].

### 7.1.1.2 FAU\_SAS

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

The TOE shall meet the requirement "Audit storage (FAU\_SAS.1)" as specified below (Common Criteria Part 2 extended).

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

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

### 7.1.2 Subset of TOE testing

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

The TOE shall meet the requirement “Subset TOE testing (FPT\_TST.2)” as specified below (Common Criteria Part 2 extended).

<b>FPT_TST.2</b>	Subset TOE testing
Hierarchical to:	No other components.
Dependencies:	No dependencies.
FPT_TST.2.1	The TSF shall run a suite of self-tests <i>at the request of the authorised user</i> to demonstrate the correct operation of the <i>alarm lines and/or following environmental sensor mechanisms</i> : <ul style="list-style-type: none"> <li>• <i>This information can be retrieved in the HRM [7].</i></li> </ul>

### 7.1.3 Memory access control

Usage of multiple applications in one Smartcard often requires code and data separation in order to prevent that one application can access code and/or data of another application. For this reason the TOE provides Area based Memory Access Control. The underlying memory management unit (MMU) is documented in section 4 of the [7].

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

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

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

The following Security Function Policy (SFP) **Memory Access Control Policy** is defined for the requirement “Security attribute based access control (FDP\_ACF.1)”:

#### **Memory Access Control Policy**

*The TOE shall control read, write, delete and execute accesses of software running at the privilege levels as defined below. Any access is controlled, regardless whether the access is on code or data or a jump on any other privilege level outside the current one.*

The memory model provides distinct, independent privilege levels separated from each other in the virtual address space. These levels are referred to as the

Infineon Technologies (IFX) level, operating system 1 and 2 levels (OS1, OS2), shared application level, and application 1 and 2 levels. A pseudo-level is the “current” level, which is simply the level on which code is currently being executed. The access rights are controlled by the MMU and related to the privilege level as explained in the HRM [7].

The TOE shall meet the requirement “Subset access control (FDP\_ACC.1)” as specified below.

<b>FDP_ACC.1</b>	Subset access control
Hierarchical to:	No other components.
Dependencies:	FDP_ACF.1 Security attribute based access control
FDP_ACC.1.1	The TSF shall enforce the <i>Memory Access Control Policy</i> on <i>all subjects (software running at the defined and assigned privilege levels), all objects (data including code stored in memories) and all the operations defined in the Memory Access Control Policy, i.e. privilege levels.</i>

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

<b>FDP_ACF.1</b>	Security attribute based access control
Hierarchical to:	No other components.
Dependencies:	FDP_ACC.1 Subset access control FMT_MSA.3 Static attribute initialization
FDP_ACF.1.1	The TSF shall enforce the <i>Memory Access Control Policy</i> to objects based on the following: <i>Subject:</i> <ul style="list-style-type: none"> <li>- <i>software running at the IFX, OS1 and OS2 privilege levels required to securely operate the chip. This includes also privilege levels running interrupt routines.</i></li> <li>- <i>software running at the privilege levels containing the application software</i></li> </ul> <i>Object:</i> <ul style="list-style-type: none"> <li>- <i>data including code stored in memories</i></li> </ul> <i>Attributes:</i> <ul style="list-style-type: none"> <li>- <i>the memory area where the access is performed to and/or</i></li> <li>- <i>the operation to be performed.</i></li> </ul>
FDP_ACF.1.2	The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: <i>evaluate the corresponding permission control information of the relevant memory range before, during or after the access so that accesses to be denied cannot be utilized by the subject attempting to perform the operation.</i>
FDP_ACF.1.3	The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: <i>none.</i>
FDP_ACF.1.4	The TSF shall explicitly deny access of subjects to objects based on the <i>following additional rules: none.</i>

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

<b>FMT_MSA.3</b>	Static attribute initialisation
Hierarchical to:	No other components.
Dependencies:	FMT_MSA.1 Management of security attributes FMT_SMR.1 Security roles
FMT_MSA.3.1	The TSF shall enforce the <i>Memory Access Control Policy</i> to provide <i>well defined</i> <sup>7</sup> default values for security attributes that are used to enforce the SFP.
FMT_MSA.3.2	The TSF shall allow <i>any subject, provided that the Memory Access Control Policy is enforced and the necessary access is therefore allowed</i> <sup>8</sup> , to specify alternative initial values to override the default values when an object or information is created.

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

<b>FMT_MSA.1</b>	Management of security attributes
------------------	-----------------------------------

<sup>7</sup> The static definition of the access rules is documented in [7]

<sup>8</sup> The Smartcard Embedded Software is intended to set the memory access control policy

Hierarchical to:	No other components.
Dependencies:	[FDP_ACC.1 Subset access control or FDP_IFC.1 Subset information flow control] FMT_SMF.1 Specification of management functions FMT_SMR.1 Security roles
FMT_MSA.1.1	The TSF shall enforce the <i>Memory Access Control Policy</i> to restrict the ability to <i>change default, modify or delete</i> the security attributes <i>permission control information to the software running on the privilege levels.</i>

The TOE shall meet the requirement “Specification of management functions (FMT\_SMF.1)” as specified below:

<b>FMT_SMF.1</b>	Specification of management functions
Hierarchical to:	No other components
Dependencies:	No dependencies
FMT_SMF.1.1	The TSF shall be capable of performing the following security management functions: <i>access the configuration registers of the MMU.</i>

### 7.1.4 Support of Cipher Schemes

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

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

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

- Advanced Encryption Standard (AES)
- Triple Data Encryption Standard (3DES)
- Elliptic Curve Cryptography (EC)
- Rivest-Shamir-Adleman (RSA)<sup>9</sup>
- Secure Hash Algorithm (SHA-2)

Note that the additional function of the EC library, providing the primitive elliptic curve operations, does not add specific security functionality.

#### 7.1.4.1 Preface regarding Security Level related to Cryptography

Preface regarding Security Level related to Cryptography

The strength of the cryptographic algorithms was not rated in the course of the product certification (see [24] Section 9, Para.4, Clause 2). But Cryptographic Functionalities with a security level of lower than 100 bits can no longer be regarded as secure without considering the application context. Therefore for these functionalities it shall be checked whether the related crypto operations are appropriate for the intended system. Some further hints and guidelines can be derived from the 'Technische Richtlinie BSI TR-02102', [www.bsi.bund.de](http://www.bsi.bund.de).

Any Cryptographic Functionality that is marked in column '*Security Level above 100 Bits*' of the following table with '*no*' achieves a security level of lower than 100 Bits (in general context).

---

<sup>9</sup> For the case the TOE comes without RSA and/or EC library, the TOE provides basic HW-related routines for RSA and/or EC calculations. For a secure library implementation the user has to implement additional countermeasures himself.



Table 19: Cryptographic TOE Functionality

Purpose	Cryptographic Function	Implemented Standard	Key Size in Bits	Security Level above 100 Bits
Key Agreement	ECDH	[23]	Key sizes corresponding to the used elliptic curves P-192, K-163 [17] and brainpoolP{160, 192}r1, brainpoolP{160, 192}t1 [18]	No
	ECDH	[23]	Key sizes corresponding to the used elliptic curves P-{224, 256, 384, 521}, K-{233, 409}, B-{233, 283, 409} [17], brainpoolP{224,256,320,384,512}r1, brainpoolP{224,256,320,384,512}t1 [18]	Yes
Cryptographic Primitive	Triple DES	[19]	k  = 112	No
	Triple DES	[19]	k  = 168	Yes
	AES	[20]	k  = 128, 192, 256	Yes
	RSA encryption / decryption / signature generation / verification (only modular exponentiation part)	[21]	Modulus length < 1976	No
	RSA encryption / decryption / signature generation / verification (only modular exponentiation part)	[21]	Modulus length = 1976 - 4096	Yes
	ECDSA signature generation / verification	[22]	Key sizes corresponding to the used elliptic curves P-192, K-163 [17] and brainpoolP{160, 192}r1, brainpoolP{160, 192}t1 [18]	No
	ECDSA signature generation / verification	[22]	Key sizes corresponding to the used elliptic curves P-{224, 256, 384, 521}, K-{233, 409}, B-{233, 283, 409} [17], brainpoolP{224,256,320,384,512}r1, brainpoolP{224,256,320,384,512}t1 [18]	Yes
	Physical True RNG PTG.2	[15]	N/A	N/A
	SHA 256 and SHA 512	[25]	None	Yes

#### 7.1.4.2 Triple-DES Operation

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

**FCS\_COP.1/DES** Cryptographic operation

Hierarchical to: No other components.

Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key management] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/DES	The TSF shall perform <i>encryption and decryption</i> in accordance with a specified cryptographic algorithm <i>Triple Data Encryption Standard (3DES)</i> with cryptographic key sizes of <i>2 x 56 bit or 3 x 56 bit</i> , that meet the following standards:  <i>National Institute of Standards and Technology (NIST), Technology Administration, U.S. Department of Data Encryption Standard (DES), NIST Special Publication 800-67, Version 1.1</i>

## Note 6:

The TOE implements the following alternative block cipher modes for the user: the Electronic Codebook Mode (ECB), the Cipher Block Chaining Mode (CBC), the Blinding Feedback Mode (BLD) and the Cipher Feedback Mode (CFB)<sup>10</sup> The BLD is described in the hardware reference manual [7] while the implementations of ECB, CBC and CFB follow the standard:

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

## Note 7:

This TOE can come with both crypto co-processors accessible, or with a blocked SCP or with a blocked Crypto2304T, or with both crypto co-processors blocked. The blocking depends on the customer demands prior to the production of the hardware. In case the SCP is blocked, no AES and 3DES computation supported by hardware is possible. In case the Crypto2304T is blocked, no RSA and EC computation supported by hardware is possible. In case of a blocked Crypto2304T the optionally delivered cryptographic and the supporting Toolbox and Base Library cannot be used in that TOE product. The use of the SHA-2 library is also possible with both crypto coprocessors blocked. No accessibility of the deselected cryptographic co-processors is without impact on any other security policy of the TOE; it is exactly equivalent to the situation where the user decides just not to use the cryptographic co-processors.

End of note.

---

<sup>10</sup> The CFB is also called Recrypt Mode.

### 7.1.4.3 AES Operation

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

<b>FCS_COP.1/AES</b>	Cryptographic operation
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/AES	The TSF shall perform <i>encryption and decryption</i> in accordance with a specified cryptographic algorithm <i>Advanced Encryption Standard (AES)</i> and cryptographic key sizes of <i>128 bit or 192 bit or 256 bit</i> that meet the following standards: <i>U.S. Department of Commerce, National Institute of Standards and Technology, Information Technology Laboratory (ITL), Advanced Encryption Standard (AES), FIPS PUB 197</i>

Note 8:

This TOE can come with both crypto co-processors accessible, or with a blocked SCP or with a blocked Crypto2304T, or with both crypto co-processors blocked. The blocking depends on the customer demands prior to the production of the hardware. In case the SCP is blocked, no AES and 3DES computation supported by hardware is possible. In case the Crypto2304T is blocked, no RSA and EC computation supported by hardware is possible. In case of a blocked Crypto2304T the optionally delivered cryptographic and the supporting Toolbox and Base Library cannot be used in that TOE product. The use of the SHA-2 library is also possible with both crypto coprocessors blocked. No accessibility of the deselected cryptographic co-processors is without impact on any other security policy of the TOE; it is exactly equivalent to the situation where the user decides just not to use the cryptographic co-processors.

Please consider also the statement of chapter 7.1.4.1.

End of note.

#### 7.1.4.4 Rivest-Shamir-Adleman (RSA) operation

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

<b>FCS_COP.1/RSA</b>	Cryptographic operation
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
FCS_COP.1.1/RSA	The TSF shall perform <i>encryption and decryption</i> in accordance with a specified cryptographic algorithm <i>Rivest-Shamir-Adleman (RSA)</i> and cryptographic key sizes <i>1024 - 4096 bits</i> that meet the following <i>standards</i>

*Encryption:*

*According to section 5.1.1 RSAEP in PKCS v2.1 RFC3447, without 5.1.1.1.*

*Decryption (with or without CRT):*

*According to section 5.1.2 RSADP in PKCS v2.1 RFC3447 for  $u = 2$ , i.e., without any  $(r_i, d_i, t_i)$ ,  $i > 2$ , therefore without 5.1.2.2.b (ii)&(v), without 5.1.2.1. 5.1.2.2.a, only supported up to  $n < 2^{2048}$*

*Signature Generation (with or without CRT)::*

*According to section 5.2.1 RSASP1 in PKCS v2.1 RFC3447 for  $u = 2$ , i.e., without any  $(r_i, d_i, t_i)$ ,  $i > 2$ , therefore without 5.2.1.2.b (ii)&(v), without 5.2.1.1. 5.2.1.2.a, only supported up to  $n < 2^{2048}$*

*Signature Verification:*

*According to section 5.2.2 RSAVP1 in PKCS v2.1 RFC3447, without 5.2.2.1.*

Please consider also the statement of chapter 7.1.4.1.

#### 7.1.4.5 Rivest-Shamir-Adleman (RSA) key generation

The key generation for the RSA shall meet the requirement “Cryptographic key generation (FCS\_CKM.1)”

<b>FCS_CKM.1/RSA</b>	Cryptographic key generation
Hierarchical to:	No other components.
Dependencies:	FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction FMT_MSA.2 Secure security attributes
FCS_CKM.1.1/RSA	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm <i>rsagen1 (PKCS v2.1 RFC3447)</i> and specified cryptographic key sizes of <i>1024 - 4096 bits</i> that meet the following <i>standard:</i>

*According to section 3.2(2) in PKCS v2.1 RFC3447,*

*for  $u=2$ , i.e., without any  $(r_i, d_i, t_i)$ ,  $i > 2$ .*

*For  $p \times q < 2^{2048}$  additionally according to section 3.2(1).*

Note 9:

For easy integration of RSA functions into the user's operating system and/or application, the library contains single cryptographic functions respectively primitives which are compliant to the standard. The primitives are referenced above. Therefore, the library supports the user to develop an application representing the standard if required.

Please consider also the statement of chapter 7.1.4.1.

End of note.

Note 10:

The TOE can be delivered with or without the RSA library. If the TOE comes with, automatically the Base Library is part of the shipment. In the case of coming without the RSA library the TOE does not provide the Additional Specific Security Functionality Rivest-Shamir-Adleman Cryptography (RSA) realized with the security functional requirements FCS\_COP.1/RSA and FCS\_CKM.1/RSA. In case of a blocked Crypto2304T the optionally delivered cryptographic and the supporting Toolbox and Base Library cannot be used in that TOE product.

End of note.

#### 7.1.4.6 Generally with regard to Elliptic Curves

The EC library is delivered as object code and in this way integrated in the user software. The certification covers the standard NIST [14] and Brainpool [15] Elliptic Curves with key lengths of 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 Bits, due to national AIS32 regulations by the BSI. Note that there are numerous other curve types, being also secure in terms of side channel attacks on this TOE, which can the user optionally add in the composition certification process.

#### 7.1.4.7 Elliptic Curve DSA (ECDSA) operation

The Modular Arithmetic Operation of the TOE shall meet the requirement "Cryptographic operation (FCS\_COP.1)" as specified below.

**FCS\_COP.1/ECDSA** Cryptographic operation

Hierarchical to: No other components.

Dependencies: [FDP\_ITC.1 Import of user data without security attributes, or  
FDP\_ITC.2 Import of user data with security attributes, or  
FCS\_CKM.1 Cryptographic key generation]  
FCS\_CKM.4 Cryptographic key destruction

FCS\_COP.1.1/ECDSA

The TSF shall perform *signature generation and signature verification* in accordance with a specified cryptographic algorithm *ECDSA* and cryptographic key sizes *160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits* that meet the following *standard*:

*Signature Generation:*

1. According to section 7.3 in ANSI X9.62 - 2005

*Not implemented is step d) and e) thereof.*

*The output of step e) has to be provided as input to our function by the caller.*

*Deviation of step c) and f):*

*The jumps to step a) were substituted by a return of the function with an error code, the jumps are emulated by another call to our function.*

2. According to sections 6.2 (6.2.2. + 6.2.3) in ISO/IEC 15946-2:2002

*Not implemented is section 6.2.1:*

*The output of 5.4.2 has to be provided by the caller as input to the function.*

*Signature Verification:*

1. According to section 7.4.1 in ANSI X9.62-2005

*Not implemented is step b) and c) thereof.*

*The output of step c) has to be provided as input to our function by*

*the caller.*  
*Deviation of step d):*  
*Beside noted calculation, our algorithm adds a random multiple of*  
*BasepointerOrder n to the calculated values u1 and u2.*

2. According to sections 6.4 (6.4.1. + 6.4.3 + 6.4.4) in ISO/IEC 15946-2:2002

*Not implemented is section 6.4.2:*

*The output of 5.4.2 has to be provided by the caller as input to the function.*

Note 11:

For easy integration of EC functions into the user’s operating system and/or application, the library contains single cryptographic functions respectively primitives which are compliant to the standard. The primitives are referenced above. Therefore, the library supports the user to develop an application representing the standard if required.

End of note.

**7.1.4.8 Elliptic Curve (EC) key generation**

The key generation for the EC shall meet the requirement “Cryptographic key generation (FCS\_CKM.1)”

<b>FCS_CKM.1/EC</b>	Cryptographic key generation
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/EC	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm <i>Elliptic Curve EC specified in ANSI X9.62-2005 and ISO/IEC 15946-1:2002</i> and specified cryptographic key sizes <i>160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits</i> that meet the following standard:  ECDSA Key Generation: 1. According to the appendix A4.3 in ANSI X9.62-2005 <i>the cofactor h is not supported.</i> 2. According to section 6.1 (not 6.1.1) in ISO/IEC 15946-1:2002

Note 12:

For easy integration of EC functions into the user’s operating system and/or application, the library contains single cryptographic functions respectively primitives which are compliant to the standard. The primitives are referenced above. Therefore, the library supports the user to develop an application representing the standard if required.

End of note.

**7.1.4.9 Elliptic Curve Diffie-Hellman (ECDH) key agreement**

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

<b>FCS_COP.1/ECDH</b>	Cryptographic operation
Hierarchical to:	No other components.
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or

FCS\_CKM.1 Cryptographic key generation]  
FCS\_CKM.4 Cryptographic key destruction

FCS\_COP.1.1/ECDH

The TSF shall perform *elliptic curve Diffie-Hellman key agreement* in accordance with a specified cryptographic algorithm *ECDH* and cryptographic key sizes *160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits* that meet the following *standard*:

1. According to section 5.4.1 in *ANSI X9.63 -2001*

*Unlike section 5.4.1.3 our, implementation not only returns the x-coordinate of the shared secret, but rather the x-coordinate and y-coordinate.*

2. According to sections 8.4.2.1, 8.4.2.2, 8.4.2.3, and 8.4.2.4 in *ISO/IEC 15946-3:2002*:

*The function enables the operations described in the four sections.*

Note 13:

The certification covers the standard NIST [14] and Brainpool [15] Elliptic Curves with key lengths of 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 Bits, due to national AIS32 regulations by the BSI. Note that there are numerous other curve types, being also secure in terms of side channel attacks on this TOE, which can the user optionally add in the composition certification process.

End of note.

Note 14:

For easy integration of EC functions into the user's operating system and/or application, the library contains single cryptographic functions respectively primitives which are compliant to the standard. The primitives are referenced above. Therefore, the library supports the user to develop an application representing the standard if required.

End of note.

Note 15:

The TOE can be delivered with or without the EC library. If the TOE comes with, automatically the Base Library is part of the shipment. In the case the TOE comes without, it does not provide the Additional Specific Security Functionality Elliptic Curve Cryptography realized with the security functional requirements FCS\_COP.1/ECDSA, FCS\_COP.1/ECDH and FCS\_CKM.1/EC. In case of a blocked Crypto2304T, the RSA and EC cryptographic library cannot be used. In case of a blocked Crypto2304T the optionally delivered cryptographic RSA and EC, as well as the supporting Toolbox and Base Library cannot be used in that TOE product.

End of note.

Note 16:

The EC primitives allow the selection of various curves. The selection of the curves depends to the user.

End of note.

#### 7.1.4.10 SHA-2 Operation

The SHA-2 Operation of the TOE shall meet the requirement "Cryptographic operation (FCS\_COP.1)" as specified below.

<b>FCS_COP.1/SHA</b>	<b>Cryptographic operation</b>
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/SHA	The TSF shall perform <i>hash-value calculation of user chosen data</i> in accordance with a specified cryptographic algorithm <i>SHA-2</i> and with cryptographic key sizes <i>of none</i> that meet the following <i>standards</i> :

*U.S. Department of Commerce / National Institute of Standards and Technology, Secure Hash Standard (SHS), FIPS PUB 180-4, 2012-March, section 6.2 SHA-256 and section 6.4 SHA-512.*

Note that the SHA-2 cryptographic operation is a keyless operation.

In case of a blocked Crypto2304T, the cryptographic libraries RSA and EC are not delivered, but the SHA library still can be part of the TOE.

Note 17:

The TOE can be delivered without the SHA-2 library. In this case the TOE does not provide the Additional Specific Security Functionality SHA-2 library, realized with the security functional requirements FCS\_COP.1/SHA.

End of note.

Note 18:

The secure hash-algorithm SHA-2 is intended to be used for signature generation, verification and generic data integrity checks. The use for keyed hash operations like HMAC or similar security critical operations involving keys, is not subject of this TOE and requires specific security improvements and DPA analysis including the operating system, which is not part of this TOE.

End of note.



### 7.1.5 Data Integrity

The TOE shall meet the requirement “Stored data integrity monitoring (FDP\_SDI.1)” as specified below:

<b>FDP_SDI.1</b>	Stored data integrity monitoring
Hierarchical to:	No other components
Dependencies:	No dependencies
FDP_SDI.1.1	The TSF shall monitor user data stored in containers controlled by the TSF for <i>inconsistencies between stored data and corresponding EDC</i> on all objects, based on the following attributes: <i>EDC value for the RAM, ROM and SOLID FLASH™ NVM.</i>

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

<b>FDP_SDI.2</b>	Stored data integrity monitoring and action
Hierarchical to:	FDP_SDI.1 stored data integrity monitoring
Dependencies:	No dependencies
FDP_SDI.2.1	The TSF shall monitor user data stored in containers controlled by the TSF for <i>data integrity and one- and/or more-bit-errors</i> on all objects, based on the following attributes: <i>corresponding EDC value for RAM, ROM and SOLID FLASH™ NVM and error correction ECC for the SOLID FLASH™ NVM.</i>
FDP_SDI.2.2	Upon detection of a data integrity error, the TSF shall <i>correct 1 bit errors in the SOLID FLASH™ NVM automatically and inform the user about more bit errors.</i>

## 7.2 TOE Security Assurance Requirements

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

Table 20: Assurance components

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

### 7.2.1 Refinements

Some refinements are taken unchanged from the PP [1]. Table 20 provides an overview where the refinement is done.

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

**Life cycle support (ALC\_CMS)**

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

**Functional Specification (ADV\_FSP)**

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

For details of the refinement see PP [1].

**7.3 Security Requirements Rationale**

**7.3.1 Rationale for the Security Functional Requirements**

The security functional requirements rationale of the TOE are defined and described in PP [1] section 6.3 for the following security functional requirements: FDP\_ITT.1, FDP\_IFC.1, FPT\_ITT.1, FPT\_PHP.3, FPT\_FLS.1, FRU\_FLT.2, FMT\_LIM.1, FMT\_LIM.2, FCS\_RNG.1, and FAU\_SAS.1.

The security functional requirements FPT\_TST.2, FDP\_ACC.1, FDP\_ACF.1, FMT\_MSA.1, FMT\_MSA.3, FMT\_SMF.1, FCS\_COP.1, FCS\_CKM.1, FDP\_SDI.1 and FDP\_SDI.2 are defined in the following description:

Table 21: Rational for additional SFR in the ST

Objective	TOE Security Functional Requirements
O.Add-Functions	<ul style="list-style-type: none"> <li>- FCS_COP.1/DES „Cryptographic operation“</li> <li>- FCS_COP.1/AES „Cryptographic operation“</li> <li>- FCS_COP.1/SHA „Cryptographic operation“</li> <li>- FCS_COP.1/RSA „Cryptographic operation“</li> <li>- FCS_COP.1/ECDSA „Cryptographic operation“</li> <li>- FCS_COP.1/ECDH „Cryptographic operation“</li> <li>- FCS_CKM.1/RSA “Cryptographic key generation”</li> <li>- FCS_CKM.1/EC „Cryptographic key generation“</li> </ul>
O.Phys-Manipulation	<ul style="list-style-type: none"> <li>- FPT_TST.2 „ Subset TOE security testing “</li> </ul>
O.Mem-Access	<ul style="list-style-type: none"> <li>- FDP_ACC.1 “Subset access control”</li> <li>- FDP_ACF.1 “Security attribute based access control”</li> <li>- FMT_MSA.3 “Static attribute initialization”</li> <li>- FMT_MSA.1 “Management of security attributes”</li> <li>- FMT_SMF.1 “Specification of Management Functions”</li> </ul>
O.Malfunction	<ul style="list-style-type: none"> <li>- FDP_SDI.1 „Stored data integrity monitoring“</li> <li>- FDP_SDI.2 „Stored data integrity monitoring and action“</li> </ul>

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

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

The security functional requirement(s) “Cryptographic operation (FCS\_COP.1)” exactly requires those functions to be implemented which are demanded by O.Add-Functions. FCS\_CKM.1/RSA supports the generation of RSA

keys, the FCS\_CKM.1/EC supports the generation of EC keys needed for this cryptographic operations. Therefore, FCS\_COP.1/RSA, FCS\_COP.1/ECDSA, FCS\_COP.1/ECDH and FCS\_CKM.1/RSA and FCS\_CKM/EC are suitable to meet the security objective. The FCS\_COP.1/SHA is a keyless algorithm and has no dependencies to FCS\_CKM.1.

The use of the supporting libraries Toolbox and Base has no impact on any security functional requirement nor does the use generate additional requirements.

The FCS\_COP.1/SHA is a keyless algorithm and has no dependencies.

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

- [FDP\_ITC.1 Import of user data without security attributes or FDP\_ITC.2 Import of user data with security attributes or FCS\_CKM.1 Cryptographic key generation],
- FCS\_CKM.4 Cryptographic key destruction,

All these requirements have to be fulfilled to support OE.Resp-Appl for FCS\_COP.1/DES (3DES algorithm) and for FCS\_COP.1/AES (AES algorithm). For the FCS\_COP.1/RSA (RSA algorithm) and FCS\_COP.1/ECDSA and FCS\_COP.1/ECDH (both EC algorithms) the FCS\_CKM.1/RSA and FCS\_CKM.1/EC are optional, since they are fulfilled by the TOE or may be fulfilled by the environment as the user can generate keys externally additionally.

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

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

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

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

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

The tested security enforcing functions are SF\_DPM Device Phase Management, SF\_CS Cryptographic Support and SF\_PMA Protection against modifying attacks.

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

The security functional requirement "Subset access control (FDP\_ACC.1)" with the related Security Function Policy (SFP) "Memory Access Control Policy" exactly require the implementation of an area based memory access control as required by O.Mem-Access. The related TOE security functional requirements FDP\_ACC.1, FDP\_ACF.1, FMT\_MSA.3, FMT\_MSA.1 and FMT\_SMF.1 cover this security objective. The implementation of these functional requirements is represented by the dedicated privilege level concept.

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

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

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

The security functional requirement “Stored data integrity monitoring (FDP\_SDI.1)” requires the implementation of an Error Detection (EDC) algorithm which detects integrity errors of the data stored in all memories. By this the malfunction of the TOE using corrupt data is prevented. Therefore FDP\_SDI.1 is suitable to meet the security objective.

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

CC part 2 defines the component FIA\_SOS.2, which is similar to FCS\_RNG.1, as follows:

FIA\_SOS.2 TSF Generation of secrets

Hierarchical to: No other components.

Dependencies: No dependencies.

- FIA\_SOS.2.1 The TSF shall provide a mechanism to generate secrets that meet [assignment: *a defined quality metric*].
- FIA\_SOS.2.2 The TSF shall be able to enforce the use of TSF generated secrets for [assignment: *list of TSF functions*].

CC part 2, annex G.3 [3], states: “This family defines requirements for mechanisms that enforce defined quality metrics on provided secrets, and generate secrets to satisfy the defined metric“. Even the operation in the element FIA\_SOS.2.2 allows listing the TSF functions using the generated secrets. Because all applications discussed in annex G.3 are related to authentication, the component FIA\_SOS.2 is also intended for authentication purposes while the term “secret” is not limited to authentication data (cf. CC part 2, paragraphs 39-42).

Paragraph 685 in [3] recommends using component FCS\_CKM.1 to address random number generation. However, this may hide the nature of the secrets used for key generation and does not allow describing random number generation for other cryptographic methods (e.g., challenges, padding), authentication (e.g., password seeds), or other purposes (e.g., blinding as a countermeasure against side channel attacks).

The component FCS\_RNG addresses general RNG including the use of but no limitation to cryptographic mechanisms. FCS\_RNG allows specifying requirements for the generation of random numbers including necessary information for the intended use. These details describe the quality of the generated data where other security services rely on. Thus by using FCS\_RNG a ST or PP author is able to express a coherent set of SFRs that include or use the generation of random numbers as a security service.

### 7.3.1.1 Dependencies of Security Functional Requirements

The dependence of security functional requirements are defined and described in PP [1] section 6.3.2 for the following security functional requirements: FDP\_ITT.1, FDP\_IFC.1, FPT\_ITT.1, FPT\_PHP.3, FPT\_FLS.1, FRU\_FLT.2, FMT\_LIM.1, FMT\_LIM.2, FCS\_RNG.1 and FAU\_SAS.1.

The dependence of security functional requirements for the security functional requirements FPT\_TST.2, FDP\_ACC.1, FDP\_ACF.1, FMT\_MSA.1, FMT\_MSA.3, FMT\_SMF.1, FCS\_COP.1, FCS\_CKM.1, FDP\_SDI.1 and FDP\_SDI.2 are defined in the following description.

Table 22: Dependency for cryptographic operation requirement

Security Functional Requirement	Dependencies	Fulfilled by security requirements
FCS_COP.1/DES	FCS_CKM.1	Yes, see comment 3
	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1) FCS_CKM.4	Yes, see comment 3
FCS_COP.1/AES	FCS_CKM.1	Yes, see comment 3
	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1) FCS_CKM.4	Yes, see comment 3
FCS_COP.1/RSA	FCS_CKM.1	Yes, see comment 3
	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1) FCS_CKM.4	Yes, see comment 3
FCS_CKM.1/RSA	FCS_CKM.2 or FCS_COP.1	Yes
	FCS_CKM.4	Yes, see comment 3
FCS_COP.1/ECDSA	FCS_CKM.1	Yes, see comment 3
	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1) FCS_CKM.4	Yes, see comment 3
FCS_CKM.1/EC	FCS_CKM.2 or FCS_COP.1	Yes
	FCS_CKM.4	Yes, see comment 3
FCS_COP.1/ECDH	FCS_CKM.1	Yes, see comment 3
	FDP_ITC.1 or FDP_ITC.2 (if not FCS_CKM.1) FCS_CKM.4	Yes, see comment 3
FCS_COP.1/SHA	No dependencies, see comment 4	Yes, see comment 3
FPT_TST.2	No dependencies, see comment 1	No, see comment 1
FDP_ACC.1	FDP_ACF.1	Yes
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Yes Yes
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	Yes Not required, see comment 2
FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1 FMT_SMR.1 FMT_SMF.1	Yes see comment 2 Yes
FMT_SMF.1	None	N/A
FDP_SDI.1	None	N/A
FDP_SDI.2	None	N/A

## Comment 1:

The TOE is already a platform representing the lowest level in a Smartcard. There is no lower or »underlying abstract machine« used by the TOE which can be tested. Therefore, the former dependency to FPT\_AMT.1 is fulfilled without further and by that dispensable. CC in the Revision 3 considered this and dropped this dependency. The requirement FPT\_TST.2 is satisfied.

End of comment.

## Comment 2:

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

End of comment.

## Comment 3:

The security functional requirement “Cryptographic operation (FCS\_COP.1)”, met by the TOE, has the following dependencies:

- [FDP\_ITC.1 Import of user data without security attributes, or
- FDP\_ITC.2 Import of user data with security attributes, or
- FCS\_CKM.1 Cryptographic key generation]
- FCS\_CKM.4 Cryptographic key destruction.

The security functional requirement “Cryptographic key management (FCS\_CKM)”, met by TOE, has the following dependencies:

- [FCS\_CKM.2 Cryptographic key distribution, or
- FCS\_COP.1 Cryptographic operation]
- FCS\_CKM.4 Cryptographic key destruction.

These requirements all address the appropriate management of cryptographic keys used by the specified cryptographic function and are not part of the PP [1]. Most requirements concerning key management shall be fulfilled by the environment since the Smartcard Embedded Software is designed for a specific application context and uses the cryptographic functions provided by the TOE.

For the security functional requirement FCS\_COP.1/DES and FCS\_COP.1/AES the respective dependencies FCS\_CKM.1, FCS\_CKM.4 and FDP\_ITC.1 or FDP\_ITC.2 have to be fulfilled by the environment. That mean, that the environment shall meet the requirements FCS\_CKM.1 and FCS\_CKM.4 as defined in [3], section 10.1 and shall meet the requirements FDP\_ITC.1 or FDP\_ITC.2 as defined in [3], section 11.7.

For the security functional requirement FCS\_COP.1/RSA, FCS\_COP.1/ECDSA and FCS\_COP.1/ECDH the respective dependencies FCS\_CKM.4 and FDP\_ITC.1 or FDP\_ITC.2 have to be fulfilled by the environment. That mean, that the environment shall meet the requirements FDP\_ITC.1 or FDP\_ITC.2 as defined in [3], section 11.7. The respective dependency FCS\_CKM.1 has to be fulfilled by the TOE with the security functional requirement FCS\_CKM.1/RSA (for FCS\_COP.1/RSA) and FCS\_CKM.1/EC (for FCS\_COP.1/ECDSA and FCS\_COP.1/ECDH) as defined in section 7.1.4. Additionally the requirement FCS\_CKM.1 can be fulfilled by the environment as defined in [3], section 10.1.

For the security functional requirement FCS\_CKM.1/RSA and FCS\_CKM.1/EC the respective dependency FCS\_COP.1 is fulfilled by the TOE. The environment covers the respective dependency FCS\_CKM.4. That mean, that the environment shall meet the requirement FCS\_CKM.4 as defined in [3], section 10.1.

The cryptographic libraries RSA, EC, SHA-2 and the Toolbox library are delivery options. If one of the libraries RSA, EC and Toolbox or combination hereof are delivered, the Base Lib is automatically part of it. Therefore the TOE may come with free combinations of or even without these libraries. In the case of coming without one or any combination of the cryptographic libraries RSA, EC and SHA-2, the TOE does not provide the Additional Specific Security Functionality Rivest-Shamir-Adleman Cryptography (RSA) and/or Elliptic Curve Cryptography (EC) and/or SHA-2. The Toolbox and Base Library are no cryptographic libraries and provide no additional specific security functionality.

In case of a blocked Crypto2304T the optionally delivered cryptographic and the supporting Toolbox and Base Library cannot be used in that TOE product. The SHA-2 library is computed in the CPUs. Therefore the IT environment has to fulfill the requirements of this chapter depending if the TOE comes with or without a/the library/ies. In case of a blocked Crypto2304T no cryptographic libraries are delivered.

End of comment.



#### Comment 4

The dependencies FCS\_CKM.1 and FMT\_CKM.4 are not required for the SHA-2 algorithm, because the SHA-2 algorithm is a keyless operation. So the environment is not obligated to meet certain requirements for key management.

End of comment.

### 7.3.2 Rationale of the Assurance Requirements

The chosen assurance level EAL5 and the augmentation with the requirements ALC\_DVS.2 and AVA\_VAN.5 were chosen in order to meet the assurance expectations explained in the following paragraphs. In Table 20 the different assurance levels are shown as well as the augmentations. The augmentations are in compliance with the Protection Profile.

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

#### ALC\_DVS.2 Sufficiency of security measures

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

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

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



**AVA\_VAN.5 Advanced methodical vulnerability analysis**

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

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

AVA\_VAN.5 has dependencies to ADV\_ARC.1 "Security architecture description", ADV\_FSP.2 "Security enforcing functional specification", ADV\_TDS.3 "Basic modular design", ADV\_IMP.1 "Implementation representation of the TSF", AGD\_OPE.1 "Operational user guidance", and AGD\_PRE.1 "Preparative procedures".

All these dependencies are satisfied by EAL5.

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

## 8 TOE Summary Specification (ASE\_TSS)

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

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

SF_DPM	Device Phase Management
SF_PS	Protection against Snooping
SF_PMA	Protection against Modification Attacks
SF_PLA	Protection against Logical Attacks
SF_CS	Cryptographic Support

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

### 8.1 SF\_DPM: Device Phase Management

The life cycle of the TOE is split-up in several phases. Chip development and production (phase 2, 3, 4) and final use (phase 4-7) is a rough split-up from TOE point of view. These phases are implemented in the TOE as test mode (phase 3) and user mode (phase 4-7). In addition a chip identification mode exists which is active in all phases. The chip identification data (O.Identification) are stored in the not changeable configuration page area of the non-volatile memory. In the same area further TOE configuration data is stored. In addition, user initialization data can be stored in the non-volatile memory during the production phase as well. During this first data programming, the TOE is still in the secure environment and in Test Mode.

The covered security functional requirement is FAU\_SAS.1 "Audit storage".

During start-up of the TOE the decision for one of the various operation modes is taken dependent on phase identifiers. The decision of accessing a certain mode is defined as phase entry protection. The phases follow also a defined and protected sequence. The sequence of the phases is protected by means of authentication.

The covered security functional requirements are FMT\_LIM.1 and FMT\_LIM.2.

During the production phase (phase 3 and 4) or after the delivery to the customer (phase 5 or phase 6), the TOE provides the possibility to download, after a successful authentication process, a user specific encryption key and user code and data into the empty (erased) SOLID FLASH™ NVM memory area as specified by the associated control information of the Flash Loader software. This process is only possible after a successful authentication process. The integrity of the loaded data is checked with a signature process. The data to be loaded may be transferred optionally in encrypted form. After finishing the load operation, the Flash Loader can be permanently deactivated, so that no further load operation with the Flash Loader is possible. These procedures are defined as phase operation limitation.

The covered security functional requirement is FPT\_LIM.2 "Limited availability".

During operation within a phase the accesses to memories are granted by the MMU controlled access rights and related privilege level.

The covered security functional requirements are FDP\_ACC.1, FDP\_ACF.1 and FMT\_MSA.1.

In addition, during each start-up of the TOE the address ranges and access rights are initialized by the STS with predefined values. The covered security functional requirement is FMT\_MSA.3.

The TOE clearly defines access rights and privilege levels in conjunction with the appropriate key management in dependency of the firmware or software to be executed. By this clearly defined management functions are implemented, enforced by the MMU, and the covered security functional requirement is FMT\_SMF.1.

During the testing phase in production within the secure environment the entire SOLID FLASH™ NVM is deleted. The covered security functional requirement is FPT\_PHP.3.

Each operation phase is protected by means of authentication and encryption. The covered security functional requirements are FDP\_ITT.1 and FPT\_ITT.1.

The **SF\_DPM** "Device Phase Management" covers the security functional requirements FAU\_SAS.1, FMT\_LIM.1, FMT\_LIM.2, FDP\_ACC.1, FDP\_ACF.1, FMT\_MSA.1, FMT\_MSA.3, FMT\_SMF.1, FPT\_PHP.3, FDP\_ITT.1 and FPT\_ITT.1.

## 8.2 SF\_PS: Protection against Snooping

All contents of all memories of the TOE are encrypted on chip to protect against data analysis on stored data as well as on internally transmitted data. There is no plain data on the chip.

In addition the data transferred over the memory bus to and from (bi-directional encryption) the CPU, Co-processor (Crypto2304T and SCP), the special SFRs and the peripheral devices (CRC, RNG and Timer) are encrypted automatically with a dynamic key change.

The memory content and bus encryption is done by the MED using a complex key management. This means that the SOLID FLASH™ NVM, RAM, CACHE and the bus are encrypted with module dedicated and dynamic keys. The only key remaining static over the product life cycle is the specific ROM key changing from customer mask to mask.

All security relevant transfer of addresses or data via the peripheral bus is dynamically masked and thus protected against readout and analysis.

No data in plain are handled anywhere on the TOE and thus also the two CPUs compute entirely masked and in addition dynamic mask changes are applied. Also the register files are masked. The symmetric cryptographic co-processor is entirely masked at any time and also here the masks change dynamically.

The CACHE being in ongoing use during operation is entirely and dynamically encrypted.

The encryption covers the data processing policy and FDP\_IFC.1 "Subset information flow control".

The covered security functional requirements are FPT\_PHP.3, FDP\_IFC.1, FPT\_ITT.1 and FDP\_ITT.1.

The user can define his own key for an SOLID FLASH™ NVM area to protect his data. This user individually chosen key is then delivered by the operating system and included in the dynamic SOLID FLASH™ NVM encryption. The user specified SOLID FLASH™ NVM area is then encrypted with his key and a dynamic component. The encryption of the memories is performed by the MED with a proprietary cryptographic algorithm and with a complex and dynamic key management providing protection against cryptographic analysis attacks. The few keys which have to be stored on the chip, for example the user chosen key and the chip specific ROM key, are protected against read out.

The covered security functional requirements are FPT\_PHP.3, FDP\_IFC.1, FPT\_ITT.1, and FDP\_ITT.1.

The CPU has no standard command set and discloses therefore no possibility for deeper analysis. The covered security functional requirement is FPT\_PHP.3.

The entire design is kept in a nonstandard way to aggravate attacks using standard analysis methods to an almost not practical condition. A smartcard dedicated CPU with a nonpublic bus protocol is used which makes analysis very complicated and time consuming. Besides the proprietary structures also the internal timing behavior is proprietary and by this aggravating significantly the analysis in addition.

Important parts of the chip are especially designed to counter leakage or side channel attacks like DPA/SPA or EMA/DEMA. Therefore, even the physical data gaining is difficult to perform, since timing and current consumption is almost independent of the processed data, protected by a bunch of other protecting means.

In the design a number of components are automatically synthesized and mixed up to disguise their physical borders and to make an analysis more difficult.

A further protective design method used is secure wiring. All security critical wires have been identified and protected by special routing measures against probing. Additionally the wires are embedded into shield lines and used as normal signal lines for operation of the chip to prevent successful probing. This measurement is called implicit shielding or short I<sup>2</sup>-shielding.

The covered security functional requirements are FPT\_PHP.3, FPT\_ITT.1 and FDP\_ITT.1.

In addition to their protection during processing of code and data their storage in the SOLID FLASH™ NVM is protected against side channel attacks too: Even if users operate with direct and static addressing for storing their secrets, the addresses are always translated to virtual addresses - if the address call is in the correct privilege level which is monitored by the MMU.

The covered security functional requirements are FPT\_PHP.3, FPT\_ITT.1 and FDP\_ITT.1.

In contrast to the linear virtual address range the physical SOLID FLASH™ NVM pages are transparently and dynamically scrambled on every page modification. This scrambling is entirely independent from the user software and the MMU. In addition a software controlled refreshing of memory pages is implemented which exchanges the physical location of a memory page by reprogramming it to another location in dependency of the performed write cycles but also including randomness. The link between the physical address and the virtual address is stored internally and is not accessible by the operating system. This measurement causes that the physical location of data is different from chip to chip even the same software may use the same virtual addresses.

A low system frequency sensor FSE is implemented to prevent the TOE from single stepping. The sensor is tested

by the user mode security life control UMSLC and connected to the clock pad. The covered security functional requirements are FPT\_PHP.3 and FPT\_FLS.1.

An induced error which cannot be corrected will be recognized by the Integrity Guard and leads to an alarm. In case of security critical detections a security alarm and reset is generated. The covered security functional requirement is FPT\_FLS.1.

The **SF\_PS** "Protection against Snooping" covers the security functional requirements FPT\_PHP.3, FDP\_IFC.1, FPT\_ITT.1, FDP\_ITT.1 and FPT\_FLS.1.

### 8.3 SF\_PMA: Protection against Modifying Attacks

First of all we can say that all security mechanisms effective against snooping **SF\_PS** apply also here since a reasonable modification of data is almost impossible on dynamically encrypted, masked, scrambled, transparently relocated, randomized and topologically protected hardware. Due to this the covered security functional requirements are FPT\_PHP.3, FDP\_IFC.1, FPT\_ITT.1, FDP\_ITT.1 and FPT\_FLS.1.

The TOE is equipped with an error detection code (EDC) which covers the memory system of RAM, ROM and SOLID FLASH™ NVM and includes also the MED, MMU and the bus system. Thus introduced failures are securely detected and in terms of single bit errors in the SOLID FLASH™ NVM also automatically corrected (FDP\_SDI.2).

In order to prevent accidental bit faults during production in the ROM, over the data stored in ROM an EDC value is calculated (FDP\_SDI.1).

The covered security functional requirements are FRU\_FLT.2, FPT\_PHP.3, FDP\_SDI.1 and FDP\_SDI.2.

If a user tears the card resulting in a power off situation during an NVM programming operation or if other perturbation is applied, no data or content loss occurs and the TOE restarts power on. The NVM tearing save write functionality covers FPT\_FLS.1 "Failure with preservation of secure state" since if the programming was not successful, the old data are still present and valid, which ensures a secure state although a programming failure occurred. This action includes also FDP\_SDI.1 "Stored data integrity monitoring" as the new data to be programmed are checked for integrity and correct programming before the page with the old data becomes the new physical page for the next new data.

The covered security functional requirement is also FPT\_PHP.3 "Resistance to physical attack", since these measures make it difficult to manipulate the write process of the NVM.

The covered security functional requirements are FPT\_FLS.1, FPT\_PHP.3 and FDP\_SDI.1.

The TOE is protected against fault and modifying attacks. The core provides the functionality of double-computing and result comparison of all tasks to detect incorrect calculations. The detection of an incorrect calculation is stored and the TOE enters a defined secure state which causes the chip internal reset process.

The implementation of two CPUs computing on the same data is by this one of the most important security features of this platform. As the results of both CPUs are compared at the end, a fault induction of modifying attacks would have to be done on both CPUs at the correct place with the correct timing – despite all other countermeasures like dynamic masking, encryption and others. As the comparison and the register files are also protected by various measures successful manipulative attacks are seen as being not practical.

During start up, the STS performs various configurations and subsystem tests. After the STS has finished, the operating system or application can call the User Mode Security Life Control (UMSLC) test. The UMSLC checks the alarm lines and/or following functions and sensors for correct operation as explained in the HRM [7].

In the case that a physical manipulation or a physical probing attack is detected, the processing of the TOE is immediately stopped and the TOE enters a secure state called security reset.

The covered security functional requirements are FPT\_FLS.1, FPT\_PHP.3 and FPT\_TST.2.

As physical effects or manipulative attacks may also address the program flow of the user software, a watchdog timer and a check point register are implemented. These features allow the user to check the correct processing time and the integrity of the program flow of the user software.

Another measure against modifying and perturbation respectively differential fault attacks (DFA) is the implementation of backward calculation in the SCP. By this induced errors are discovered.

The covered security functional requirements are FPT\_FLS.1, FDP\_IFC.1, FPT\_ITT.1, FDP\_ITT.1 and FPT\_PHP.3.

The RMS provides the user also the testing of all security features enabled to generate an alarm. This security testing is called user mode security life control (UMSLC). As attempts to modify the security features will be detected from the test, the covered security functional requirement is FPT\_TST.2.

All communication via the busses is in addition protected by a monitored hardware handshake. If the handshake was not successful an alarm is generated.

The covered security functional requirements are FPT\_FLS.1 and FPT\_PHP.3.

The virtual memory system and privilege level model are enforced by the MMU. This controls the access rights throughout the TOE. There is a clear differentiation within the privilege levels defined. The covered security functional requirements are FDP\_ACC.1, FDP\_ACF.1, FMT\_MSA.1, FMT\_MSA.3 and FMT\_SMF1.

The **SF\_PMA** "Protection against Modifying Attacks" covers the security functional requirements FPT\_PHP.3, FDP\_IFC.1, FPT\_ITT.1, FDP\_ITT.1, FMT\_MSA.1, FMT\_MSA.3, FMT\_SMF.1, FDP\_ACC.1, FDP\_ACF.1, FRU\_FLT.2, FPT\_TST.2, FDP\_SDI.1, FDP\_SDI.2 and FPT\_FLS.1.

## 8.4 SF\_PLA: Protection against Logical Attacks

The memory access control of the TOE uses a memory management unit (MMU) to control the access to the available physical memory by using virtual memory addresses and to segregate the code and data to a privilege level model. The MMU controls the address permissions of up seven privileged levels and gives the software the possibility to define different access rights for the privileged levels 3 to 7. The address permissions of the privilege levels are controlled by the MMU. In case of an access violation the MMU will trigger a reset and then a trap service routine can react on the access violation. The policy of setting up the MMU and specifying the memory ranges for the privilege levels – with the exception of the IFX level - is defined from the user software (OS). The privilege levels 0, 1 and 2 are reserved for TOE internal operations. The privilege levels 3 and 4 are reserved for operation systems and the privilege levels 5, 6 and 7 are reserved for applications.

As the TOE provides support for separation of memory areas the covered security functional requirements are FDP\_ACC.1 “Subset access control”, FDP\_ACF.1 “Security attribute based access control”, FMT\_MSA.3 “Static attribute initialization”, FMT\_MSA.1 “Management of security attributes” and FMT\_SMF.1 “Specification of Management functions”.

The TOE provides the possibility to protect the property rights of user code and data by the encryption of the SOLID FLASH™ NVM memory areas with a specific key defined by the user. Due to this key management FDP\_ACF.1 is fulfilled. In addition, all memories present on the TOE are individually encrypted using individual keys assigned by complex key management. All data are protected by means of encryption or masking also during transportation via the busses. Induced errors are recognized by the Integrity Guard concept and lead to an alarm. In case of security critical errors a security alarm is generated and the TOE ends up in a secure state. The covered security functional requirements are FPT\_PHP.3, FDP\_ITT.1, FDP\_IFC.1 and FPT\_FLS.1.

Beside the access protection and key management, also the use of illegal operation code is detected and will release a security reset.

The **SF\_PLA** “Protection against Logical Attacks” covers the security functional requirements FDP\_ACC.1, FDP\_ACF.1, FMT\_MSA.1, FMT\_MSA.3, FPT\_PHP.3, FDP\_ITT.1, FDP\_IFC.1, FPT\_FLS.1 and FMT\_SMF.1.

## 8.5 SF\_CS: Cryptographic Support

The TOE is equipped with a hardware accelerator for TDES and AES cryptographic operations. This security function is introduced to include the cryptographic operation in the scope of the evaluation as the cryptographic function respectively mathematic algorithm itself is not used from the TOE security policy.

On the other hand these functions are of special interest for the use of the hardware as platform for the software. The components are a co-processor supporting the DES and AES algorithms and a combination of a co-processor and software modules to support RSA cryptography, ECDSA signature generation and verification, ECDH key agreement and EC public key calculation and public key testing. The RSA cryptographic key generation is not supported for this TOE.

Note that the additional function of the EC library, ECC\_ADD, providing the primitive elliptic curve operations, does not add specific security functionality and that the according user guidance abbreviates the Elliptic Curve cryptographic functions with ECC.

### 8.5.1 3DES

The TOE supports the encryption and decryption in accordance with the specified cryptographic algorithm Triple Data Encryption Standard (3DES) with cryptographic key sizes of 112 bit or 168 bit meeting the standard:

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

The TOE implements the following alternative block cipher modes for the user: the Electronic Codebook Mode (ECB), the Cipher Block Chaining Mode (CBC), the Blinding Feedback Mode (BLD) and the Cipher Feedback Mode (CFB). The BLD is described in the hardware reference manual [7] while the implementations of ECB, CBC and CFB follow the standard:

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

Please consider also the statement of chapter 7.1.4.1.

The covered security functional requirements are FCS\_COP.1/DES.

### 8.5.2 AES

The TOE supports the encryption and decryption in accordance with the specified cryptographic algorithm Advanced Encryption Standard (AES) and cryptographic key sizes of 128 bit or 192 bit or 256 bit that meet the standard:

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

Please consider also the statement of chapter 7.1.4.1.

The covered security functional requirement is FCS\_COP.1/AES.

### 8.5.3 RSA

#### Encryption, Decryption, Signature Generation and Verification

The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm Rivest-Shamir-Adleman (RSA) and cryptographic key sizes 1024 - 4096 bits that meet the following standards

*Encryption:  
According to section 5.1.1 RSAEP in PKCS v2.1 RFC3447,  
without 5.1.1.1.*

*Decryption (with or without CRT):  
According to section 5.1.2 RSADP in PKCS v2.1 RFC3447  
for  $u = 2$ , i.e., without any  $(r_i, d_i, t_i)$ ,  $i > 2$ ,  
therefore without 5.1.2.2.b (ii)&(v), without 5.1.2.1.  
5.1.2.2.a, only supported up to  $n < 2^{2048}$*

*Signature Generation (with or without CRT)::  
According to section 5.2.1 RSASP1 in PKCS v2.1 RFC3447  
for  $u = 2$ , i.e., without any  $(r_i, d_i, t_i)$ ,  $i > 2$ ,  
therefore without 5.2.1.2.b (ii)&(v), without 5.2.1.1.  
5.2.1.2.a, only supported up to  $n < 2^{2048}$*

*Signature Verification:  
According to section 5.2.2 RSAVP1 in PKCS v2.1 RFC3447,  
without 5.2.2.1.*

Please consider also the statement of chapter 7.1.4.1.

#### Asymmetric Key Generation



The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *RSA specified in PKCS#1 v2.1* and specified cryptographic key sizes of *1024 – 4096 bits* that meet the following standard:

*According to section 3.2(2) in PKCS v2.1 RFC3447,  
for  $u=2$ , i.e., without any  $(r_i, d_i, t_i)$ ,  $i > 2$ .  
For  $p \times q < 2^{2048}$  additionally according to section 3.2(1).*

Note 19:

For easy integration of RSA functions into the user's operating system and/or application, the library contains single cryptographic functions respectively primitives which are compliant to the standard. The primitives are referenced above. Therefore, the library supports the user to develop an application representing the standard if required.

End of note.

The covered security functional requirement is FCS\_COP.1/RSA and FCS\_CKM.1/RSA.

### 8.5.4 Elliptic Curves

The certification covers the standard NIST [14] and Brainpool [15] Elliptic Curves with key lengths of 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 Bits, due to national AIS32 regulations by the BSI. Note that numerous other side channel attack resistant curve types exist, which the user optionally can add in the composition certification process.

#### Signature Generation and Verification

The TSF shall perform signature generation and signature verification in accordance with a specified cryptographic algorithm ECDSA and cryptographic key sizes *160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits* that meet the following standard:

*Signature Generation:*

- 1. According to section 7.3 in ANSI X9.62 - 2005  
Not implemented is step d) and e) thereof.  
The output of step e) has to be provided as input to our function by the caller.  
Deviation of step c) and f):  
The jumps to step a) were substituted by a return of the function with an error code, the jumps are emulated by another call to our function.*
- 2. According to sections 6.2 (6.2.2. + 6.2.3) in ISO/IEC 15946-2:2002  
Not implemented is section 6.2.1:  
The output of 5.4.2 has to be provided by the caller as input to the function.*

*Signature Verification:*

- 1. According to section 7.4.1 in ANSI X9.62–2005  
Not implemented is step b) and c) thereof.  
The output of step c) has to be provided as input to our function by the caller.  
Deviation of step d):  
Beside noted calculation, our algorithm adds a random multiple of BasepointerOrder  $n$  to the calculated values  $u1$  and  $u2$ .*
- 2. According to sections 6.4 (6.4.1. + 6.4.3 + 6.4.4) in ISO/IEC 15946-2:2002  
Not implemented is section 6.4.2:  
The output of 5.4.2 has to be provided by the caller as input to the*



*function.*

### Asymmetric Key Generation

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm Elliptic Curve EC specified in ANSI X9.62-1998 and ISO/IEC 15946-1:2002 and specified cryptographic key sizes 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits that meet the following standard:

ECDSA Key Generation:

1. According to the appendix A4.3 in ANSI X9.62-2005 the cofactor  $h$  is not supported.
2. According to section 6.1 (not 6.1.1) in ISO/IEC 15946-1:2002

### Asymmetric Key Agreement

The TSF shall perform elliptic curve Diffie-Hellman key agreement in accordance with a specified cryptographic algorithm ECDH and cryptographic key sizes 160, 163, 192, 224, 233, 256, 283, 320, 384, 409, 512 or 521 bits that meet the following standard:

1. According to section 5.4.1 in ANSI X9.63 -2001

*Unlike section 5.4.1.3 our implementation not only returns the  $x$ -coordinate of the shared secret, but rather the  $x$ -coordinate and  $y$ -coordinate.*

2. According to sections 8.4.2.1, 8.4.2.2, 8.4.2.3, and 8.4.2.4 in

*ISO/IEC 15946-3:2002:*

*The function enables the operations described in the four sections.*

Note 20:

For easy integration of EC functions into the user's operating system and/or application, the library contains single cryptographic functions respectively primitives which are compliant to the standard. The primitives are referenced above. Therefore, the library supports the user to develop an application representing the standard if required.

End of note.

The covered security functional requirements are FCS\_COP.1/ECDSA, FCS\_CKM.1/EC and FCS\_COP.1/ECDH.

### 8.5.5 SHA-2

The TOE comes optionally with the SHA-2 library for hash value calculation. Regarding the SHA-2 library it has to be noted that the secure hash-algorithm SHA-2 is intended to be used for signature generation, verification and generic data integrity checks. The use for keyed hash operations like HMAC or similar security critical operations involving keys, is not subject of this TOE and requires specific security improvements and DPA analysis including the operating system, which is not part of this TOE. Nevertheless, following is valid:

The TSF shall perform hash-value calculation of user chosen data in accordance with a specified cryptographic algorithm SHA-2 and with cryptographic key sizes of none that meet the following standards:

*U.S. Department of Commerce / National Institute of Standards and Technology, Secure Hash Standard (SHS), FIPS PUB 180-4, 2012-March, section 6.2 SHA-256 and section 6.4 SHA-512.*

The covered security functional requirement is FCS\_COP.1/SHA.

### 8.5.6 Toolbox Library

The toolbox provides the following basic long integer arithmetic and modular functions in software, supported by the cryptographic coprocessor: Addition, subtraction, division, multiplication, comparison, reduction, modular addition, modular subtraction, modular multiplication, modular inversion and modular exponentiation. No security relevant policy, mechanism or function is supported. The toolbox library is deemed for software developers as support for simplified implementation of long integer and modular arithmetic operations.

The toolbox does not cover security functional requirements.

### 8.5.7 Base Library

The Base Library provides the low level interface to the asymmetric cryptographic coprocessor and has no user available interface. The base library does not provide any security functionality, implements no security mechanism, and does not provide additional specific security functionality.

The Base Library does not cover security functional requirements and has no user interface.

### 8.5.8 PTRNG respectively TRNG

Random data is essential for cryptography as well as for security mechanisms. The TOE is equipped with a physical True Random Number Generator (PTRNG or TRNG, FCS\_RNG.1). The random data can be used from the Smartcard Embedded Software and is also used from the security features of the TOE, like masking. The TRNG implements also self-testing features. The PTRNG or TRNG meets the requirements of the functionality class PTG.2 of AIS31 [6].

The covered security functional requirement is FCS\_RNG.1, FPT\_PHP.3, FDP\_ITT.1, FPT\_ITT.1, FPT\_TST.2 and FPT\_FLS.1.

The **SF\_CS** "Cryptographic Support" covers the security functional requirements FCS\_COP.1/DES, FCS\_COP.1/AES, FCS\_COP.1/RSA, FCS\_CKM.1/RSA, FCS\_COP.1/ECDSA, FCS\_CKM.1/EC, FCS\_COP.1/ECDH, FCS\_COP.1/SHA, FPT\_PHP.3, FDP\_ITT.1, FPT\_ITT.1, FPT\_TST.2, FPT\_FLS.1 and FCS\_RNG.1.

Note 21:

The cryptographic libraries RSA, EC, SHA-2 and the Toolbox library are delivery options. If one of the libraries RSA, EC and Toolbox or combination hereof are delivered, the Base Lib is automatically part of it. Therefore the TOE may come with free combinations of or even without these libraries. In the case of coming without one or any combination of the cryptographic libraries RSA, EC and SHA-2, the TOE does not provide the Additional Specific Security Functionality Rivest-Shamir-Adleman Cryptography (RSA) and/or Elliptic Curve Cryptography (EC) and/or SHA-2. The Toolbox and Base Library are no cryptographic libraries and provide no additional specific security functionality.

End of note.

Note 22:

This TOE can come with both crypto co-processors accessible, or with a blocked SCP or with a blocked

Crypto2304T, or with both crypto co-processors blocked. The blocking depends on the customer demands prior to the production of the hardware. In case the SCP is blocked, no AES and 3DES computation supported by hardware is possible. In case the Crypto2304T is blocked, no RSA and EC computation supported by hardware is possible. The use of the SHA-2 library is also possible with both crypto coprocessors blocked. No accessibility of the deselected cryptographic co-processors is without impact on any other security policy of the TOE; it is exactly equivalent to the situation where the user decides just not to use the cryptographic co-processors.  
End of note.

## **8.6 Assignment of Security Functional Requirements to TOE's Security Functionality**

The justification and overview of the mapping between security functional requirements (SFR) and the TOE's security functionality (SF) is given in sections the sections above. The results are shown in Table 23. The security functional requirements are addressed by at least one relating security feature.

The various functional requirements are often covered manifold. As described above the requirements ensure that the TOE is checked for correct operating conditions and if a not correctable failure occurs that a stored secure state is achieved, accompanied by data integrity monitoring and actions to maintain the integrity although failures occurred. An overview is given in following table:

Table 23: Mapping of SFR and SF

Security Functional Requirement	SF_DPM	SF_PS	SF_PMA	SF_PLA	SF_CS
FAU_SAS.1	X				
FMT_LIM.1	X				
FMT_LIM.2	X				
FDP_ACC.1	X		X	X	
FDP_ACF.1	X		X	X	
FPT_PHP.3	X	X	X	X	X
FDP_ITT.1	X	X	X	X	X
FDP_SDI.1			X		
FDP_SDI.2			X		
FDP_IFC.1		X	X	X	
FMT_MSA.1	X		X	X	
FMT_MSA.3	X		X	X	
FMT_SMF.1	X		X	X	
FRU_FLT.2			X		
FPT_ITT.1	X	X	X		X
FPT_TST.2			X		X
FPT_FLS.1		X	X	X	X
FCS_RNG.1					X
FCS_COP.1/DES					X
FCS_COP.1/AES					X
FCS_COP.1/RSA					X
FCS_COP.1/ECDSA					X
FCS_COP.1/ECDH					X
FCS_COP.1/SHA					X
FCS_CKM.1/RSA					X
FCS_CKM.1/EC					X

## 8.7 Security Requirements are internally Consistent

For this chapter the PP [1] section 6.3.4 can be applied completely.

In addition to the discussion in section 6.3 of PP [1] the security functional requirement FCS\_COP.1 is introduced. The security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced also protect the cryptographic algorithms implemented according to the security functional requirement FCS\_COP.1. Therefore, these security functional requirements support the secure implementation and operation of FCS\_COP.1.

As disturbing, manipulating during or forcing the results of the test checking the security functions after TOE delivery, this security functional requirement FPT\_TST.2 has to be protected. An attacker could aim to switch off or disturb certain sensors or filters and preserve the detection of his manipulation by blocking the correct operation of FPT\_TST.2. The security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced also protect the security functional requirement FPT\_TST.2. Therefore, the related security functional requirements support the secure implementation and operation of FPT\_TST.2.

The requirement FPT\_TST.2 allows testing of some security mechanisms by the Smartcard Embedded Software after delivery. In addition, the TOE provides an automated continuous user transparent testing of certain functions.

The implemented privilege level concept represents the area based memory access protection enforced by the MMU. As an attacker could attempt to manipulate the privilege level definition as defined and present in the TOE, the functional requirement FDP\_ACC.1 and the related other requirements have to be protected themselves. The security functional requirements required to meet the security objectives O.Leak-Inherent, O.Phys-Probing, O.Malfunction, O.Phys-Manipulation and O.Leak-Forced also protect the area based memory access control function implemented according to the security functional requirement described in the security functional requirement FDP\_ACC.1 with reference to the Memory Access Control Policy and details given in FDP\_ACF.1. Therefore, those security functional requirements support the secure implementation and operation of FDP\_ACF.1 with its dependent security functional requirements.

The requirement FDP\_SDI.2.1 allows detection of integrity errors of data stored in memory. FDP\_SDI.2.2 in addition allows correction of one bit errors or taking further action. Both meet the security objective O.Malfunction. The requirements FRU\_FLT.2, FPT\_FLS.1, and FDP\_ACC.1 which also meet this objective are independent from FDP\_SDI.2 since they deal with the observation of the correct operation of the TOE and not with the memory content directly.

## 9 References

### 9.1 Literature

- [1] Security IC Platform Protection Profile, Version 1.0, 15.06.2007, BSI-PP-0035
- [2] Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model; Version 3.1, Revision 5, April 2017, CCMB-2017-04-001
- [3] Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Requirements; Version 3.1, Revision 5, April 2017, CCMB-2017-04-002
- [4] Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Requirements; Version 3.1, Revision 5 April 2017, CCMB-2017-04-003
- [5] Status report, List of all available user guidance
- [6] Functionality classes and evaluation methodology for physical random number generators AIS31, Version 3.0, 05.15.2013
- [7] SLx 70 Family – Hardware Reference Manual,, Infineon Technologies AG, as referenced in Table 1: Identification
- [11] Joint Interpretation Library, Application of Attack Potential to Smartcards, Version 2.9, January 2013
- [12] M7801 / M7820 Controller Family for Security Applications, Errata Sheet, as referenced in Table 1: Identification
- [13] SLE 70 Family Programmer's Reference User's Manual, Infineon Technologies AG, as referenced in Table 1: Identification
- [14] NIST: FIPS publication 186-4: Digital Signature Standard (DSS), September 2013
- [15] IETF: RFC 5639, Elliptic Curve Cryptography (ECC) Brainpool Standard Curves and Curve Generation, March 2010, <http://www.ietf.org/rfc/rfc5639.txt>
- [19] NIST Special Publication 800-67, Revision 1, 2012-01
- [20] SP 800-38A, Recommendations for Block Cipher Modes of Operation, Methods and Techniques, December 2001
- [21] PKCS #1: RSA Cryptography Standard, v2.1, June 14, 2002, RSA Laboratories
- [22] American National Standard for Financial Services ANS X9.62-2005, Public Key Cryptography for the Financial Services Industry, The Elliptic Curve Digital Signature Algorithm (ECDSA), November 16, 2005, American National Standards Institute
- [23] American National Standard for Financial Services X9.63-2001, Public Key Cryptography for the Financial Services Industry: Key Agreement and Key Transport Using Elliptic Curve Cryptography, November 20, 2001, American National Standards Institute
- [24] Act on the Federal Office for Information Security (BSI-Gesetz – BSIG), Bundesgesetzblatt I p. 2821, 2009-08-14
- [25] U.S. Department of Commerce, National Institute of Standards and Technology, Secure Hash Standard (SHS), FIPS PUB 180-4, section 6.2 SHA-256 and section 6.4 SHA-512

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

## 10 Appendix

In Table 21 the hash signatures of the respective CL70 Crypto Library file are documented. For convenience purpose several hash values are referenced.

Table 24: Reference hash values of the CL70 Crypto Libraries

### **RSA, EC, Toolbox Version v1.02.013**

CI70-LIB-base-XSMALL-HUGE.lib:

MD5=d080392bc14a65de9094d846498f28a3

SHA1=4149318953b22876c6d9f712e084f00dccaac88f

SHA256=c08bf0778baf3a25123e1ff45590eeec6bff29cb38ede2a07a377f968d5eeade

CI70-LIB-2k-XSMALL-HUGE.lib:

MD5=e1829fa50cbd46f912e40528e92e77d4

SHA1=41a4c013fe08cbcf4917753076d8c035657040a0

SHA256=afe2dc4b3eceb6d67dab8add2581f3ceb4f6268d5f6c0091f7d975afbbec86ca

CI70-LIB-4k-XSMALL-HUGE.lib:

MD5=3c2ac3030c2abbc9e6d32b46c244f59b

SHA1=0867b74168c2b228a12c2835de92262d9536bdde

SHA256=11736a910bdc9e8a2d74b56db60d2002ff8bd9ba49be8c8fc08d744128ac6e3b

CI70-LIB-ecc-XSMALL-HUGE.lib:

MD5=9ccf23232e16645448323670e8fa3171

SHA1=315952fc79e4e711e6f95e2b9d547a5c91d88d1c

SHA256=69ad0d5bfaf2308c24d19ee8824d61952a73c273dd57ee19612dace6ba92e772

CI70-LIB-toolbox-XSMALL-HUGE.lib:

MD5=4c577bcf9853c8c030b84ebe19d22b8d

SHA1=4c12dc67dad4bbe88c4b23c43a275b3ad3be71f3

SHA256=e6de94b27ffce43b8a023c04ceb86795585617b4cac8e7a53a78cf273b1fe8fc

**SHA-2 Library Version 1.01:**

SHA-2 values computed from: SLE70-SHA2-Lib\_RE\_1v01\_2009-06-29.LIB

MD5=70d2df490185b419fb820d597d82d117

SHA1= df15ff79b5f5ab70bbad0ee031953e1877cabd47

SHA256=765fc5d47cf8274833476406b24010a56ebcfd4b0972704ddd27e2d3e3e086f8



## 11 List of Abbreviations

AES	Advanced Encryption Standard
AIS31	“Anwendungshinweise und Interpretationen zu ITSEC und CC Funktionalitätsklassen und Evaluationsmethodologie für physikalische Zufallszahlengeneratoren”
API	Application Programming Interface
CC	Common Criteria
CI	Chip Identification Mode (STS-CI)
CIM	Chip Identification Mode (STS-CI), same as CI
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
Crypto2304T	Asymmetric Cryptographic Processor
CRT	Chinese Remainder Theorem
DPA	Differential Power Analysis
DFA	Differential Failure Analysis
DRNG	Deterministic Random Noise Generator
EC	Elliptic Curve
ECC	Error Correction Code
EDC	Error Detection Code
EDU	Error Detection Unit
EEPROM	Electrically Erasable and Programmable Read Only Memory
EMA	Electromagnetic analysis
SOLID FLASH™ NVM	Infineon Technologies nonvolatile memory
HW	Hardware
IC	Integrated Circuit
ICO	Internal Clock Oscillator
ID	Identification
IMM	Interface Management Module
ITP	Interrupt and Peripheral Event Channel Controller
I/O	Input/Output
IRAM	Internal Random Access Memory
ITSEC	Information Technology Security Evaluation Criteria
M	Mechanism
MED	Memory Encryption and Decryption
MMU	Memory Management Unit
O	Object
OS	Operating system
PEC	Peripheral Event Channel
PRNG	Pseudo Random Number Generator
PROM	Programmable Read Only Memory
PTRNG	Physical True Random Noise Generator

---

RAM	Random Access Memory
RMS	Resource Management System
RNG	Random Number Generator
ROM	Read Only Memory
RSA	Rives-Shamir-Adleman Algorithm
SAM	Service Algorithm Minimal
SCP	Symmetric Cryptographic Processor
SF	Security Feature
SFR	Special Function Register, as well as Security Functional Requirement
	The specific meaning is given in the context
SPA	Simple power analysis
STS	Self-Test Software
SW	Software
SO	Security objective
T	Threat
TM	Test Mode (STS)
TOE	Target of Evaluation
TRNG	True Random Number Generator
TSC	TOE Security Functions Control
TSF	TOE Security Functionality
UART	Universal Asynchronous Receiver/Transmitter
UM	User Mode (STS)
UmSLC	User mode Security Life Control
WDT	Watch Dog Timer
XRAM	eXtended Random Access Memory
3DES	Triple DES Encryption Standard

## 12 Glossary

Application Program/Data	Software which implements the actual TOE functionality provided for the user or the data required for that purpose
Central Processing Unit	Logic circuitry for digital information processing
Chip	Integrated Circuit]
Chip Identification Data	Data stored in the SOLID FLASH™ NVM containing the chip type, lot number (including the production site), die position on wafer and production week and data stored in the ROM containing the STS version number
Chip Identification Mode	Operational status phase of the TOE, in which actions for identifying the individual chip by transmitting the Chip Identification Data take place
Controller	IC with integrated memory, CPU and peripheral devices
Crypto2304T	Cryptographic coprocessor for asymmetric cryptographic operations (RSA, Elliptic Curves)
Cyclic Redundancy Check	Process for calculating checksums for error detection
Electrically Erasable and Programmable Read Only Memory (EEPROM)	Non-volatile memory permitting electrical read and write operations with Infineon trademark SOLID FLASH™ NVM
End User	Person in contact with a TOE who makes use of its operational capability
Firmware	Part of the software implemented as hardware
Hardware	Physically present part of a functional system (item)
Integrated Circuit	Component comprising several electronic circuits implemented in a highly miniaturized device using semiconductor technology
Internal Random Access Memory	RAM integrated in the CPU
Mechanism	Logic or algorithm which implements a specific security function in hardware or software
Memory Encryption and Decryption	Method of encoding/decoding data transfer between CPU and memory
Memory	Hardware part containing digital information (binary data)
Microprocessor	CPU with peripherals
Object	Physical or non-physical part of a system which contains information and is acted upon by subjects
Operating System	Software which implements the basic TOE actions necessary for operation
Programmable Read Only Memory	Non-volatile memory which can be written once and then only permits read operations
Random Access Memory	Volatile memory which permits write and read operations
Random Number Generator	Hardware part for generating random numbers
Read Only Memory	Non-volatile memory which permits read operations only
Resource Management System	Part of the firmware containing SOLID FLASH™ NVM programming routines, AIS31 testbench etc.
SCP	Symmetric cryptographic coprocessor for symmetric cryptographic operations (3DES, AES).

Self Test Software	Part of the firmware with routines for controlling the operating state and testing the TOE hardware
Security Function	Part(s) of the TOE used to implement part(s) of the security objectives
Security Target	Description of the intended state for countering threats
Smart Card	Plastic card in credit card format with built-in chip
Software	Information (non-physical part of the system) which is required to implement functionality in conjunction with the hardware (program code)
Subject	Entity, generally in the form of a person, who performs actions
Target of Evaluation	Product or system which is being subjected to an evaluation
Test Mode	Operational status phase of the TOE in which actions to test the TOE hardware take place
Threat	Action or event that might prejudice security
User Mode	Operational status phase of the TOE in which actions intended for the user takes place