Bayer Innovation

escrypt
Embedded Security

Security Target Lite

PhenoStor® Card Reader GRE100010

Produced and sponsored by

Bayer Innovation GmbH

Prepared by

escrypt GmbH - Embedded Security

According to the

Common Criteria for Information Technology

Security Evaluation (CC) 2.2

at Level EAL 3

Version:	1.1
Status:	Public
Date:	8-Mar-07

Change History

Version	*Author*	*Comments*	*Date*
1.0	escrypt	Created document	08-Mar-07
1.1	escrypt	Minor modifications	08-Mar-07

Table of contents

List of figures

List of tables

1 Introduction

1.1 ST identification

This Security Target Lite (ST) for PhenoStor® Card Reader GRE100010 (referred to as 'card reader' in the following) has the following identification:

- phenostor-stl-1.1-070308,

and was issued on 08-Mar-07.

The ST refers to the PhenoStor® Card Reader GRE100010 which is a secure holographic memory card reader (TOE) provided by Bayer Innovation GmbH for a Common Criteria evaluation and whose security relevant part was developed by escrypt GmbH – Embedded Security.

1.2 ST overview

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

- PhenoStor® Card Reader GRE100010.

The TOE is a card reader device for holographic memory cards providing the functionality of reading data from a holographic storage card and securely transmitting the data to the interface unit connected to the card reader.

The card reader system reads and processes information from a special holographic memory card which contains sensitive data that is symmetrically encrypted and MAC-protected or symmetrically encrypted and digitally signed. Additionally, the data on the holographic memory card can be analogue scrambled. These operations are performed by the card writer (the CW – **C**ard **W**riter) in the IT environment of the TOE. The CW can be realized as a set of different software and hardware components that need not to form a physical whole and may be topologically distributed. The TOE itself enables securely transmitting data from a holographic memory card to peripheral end devices over the interface unit (the IU – **I**nterface **U**nit – and the peripheral devices are out of scope of the evaluation) using cryptographically protected external communication interfaces. All the cryptographic keys on the TOE can be updated by the administrator (the AU - **A**dministrator **U**nit) at any time in the end user phase. The security of the card reader is based upon the STMicroelectronics ST19WP18-D which is a smartcard integrated circuit with its dedicated software certified in compliance with Common Criteria 2.2 [2-4], EAL5 augmented with ALC_DVS.2, AVA_MSU.3 and AVA_VLA.4 and in accordance with Smartcard IC Platform Protection Profile BSI-PP-0002-2001 [6] and Protection Profile

Smartcard Integrated Circuit PP/9806 [7]. The cryptographic functionality of the certified cryptographic library (3DES, RSA, SHA-1) of ST19WP18-D is used for protecting the communication within the TOE, and for the secure communication channel between the TOE and IU and between the TOE and the administrator's device, respectively.

1.3 CC conformance

The evaluation is based upon:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 2.2, January 2004 [2]
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 2.2, January 2004 [3]
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 2.2, January 2004 [4]
- Common Methodology for Information Technology Security Evaluation, Evaluation Methodology; Version 2.2, January 2004 [5]

The ST claims the following Common Criteria conformances:

- Part 2 extended (Note: The supplement "extended" only refers to two additional SFR components.),
- Part 3 conformant,
- EAL 3.

The selected assurance level is EAL3. The minimum strength level for the TOE security functions is SOF-medium (Strength of functions medium).

In order to avoid redundancy and to minimize the evaluation efforts, the evaluation of the TOE will be conducted as a composite evaluation and will make use of the evaluation results of the CC evaluation of the underlying smartcard IC ST19WP18-D provided by STMicroelectronics.

2 TOE description

2.1 TOE definition

The TOE is a holographic memory card reader. The TOE basically consists of an optical module for reading out data from a holographic memory card, a standard microcontroller with dedicated software for the main control of the card reader as well as the ST19WP18-D security controller with its dedicated software and embedded software running on ST19WP18-D.

The TOE physically reads data from a PhenoStor® holographic card [1] and sends the data to the IU, which is out of the evaluation scope. The IU decides which data is needed for the application. If this data is cryptographically protected, it is sent to the TOE where it is decrypted and its signature (or MAC) is verified. Then the data is sent to the IU through a secure communication channel. Now the data can be used by the application.

The functionalities of the card reader and card writer are completely physically and logically separated, and the TOE comprises the PhenoStor® holographic memory card reader only.

The embedded software running on the security controller is the main object of the evaluation, since all security relevant features of the TOE are implemented on its basis and on the basis of the security controller ST19WP18-D with its dedicated software.

2.2 TOE hardware description

The hardware of the TOE consists of the following physical components (Figure 1):

- ST19WP18-D by STMicroelectronics as the secure computational base of the card reader;
- AVR ATmega128 standard microcontroller providing the functionality of logically integrating the components of the card reader system and the communication features;
- Optical module for physically reading data from holographic memory cards;
- Ethernet interface;
- Serial port interface;
- Keyboard, display;

- Non-intelligent hardware components including data transmission and power supply wires, capacitors, etc.

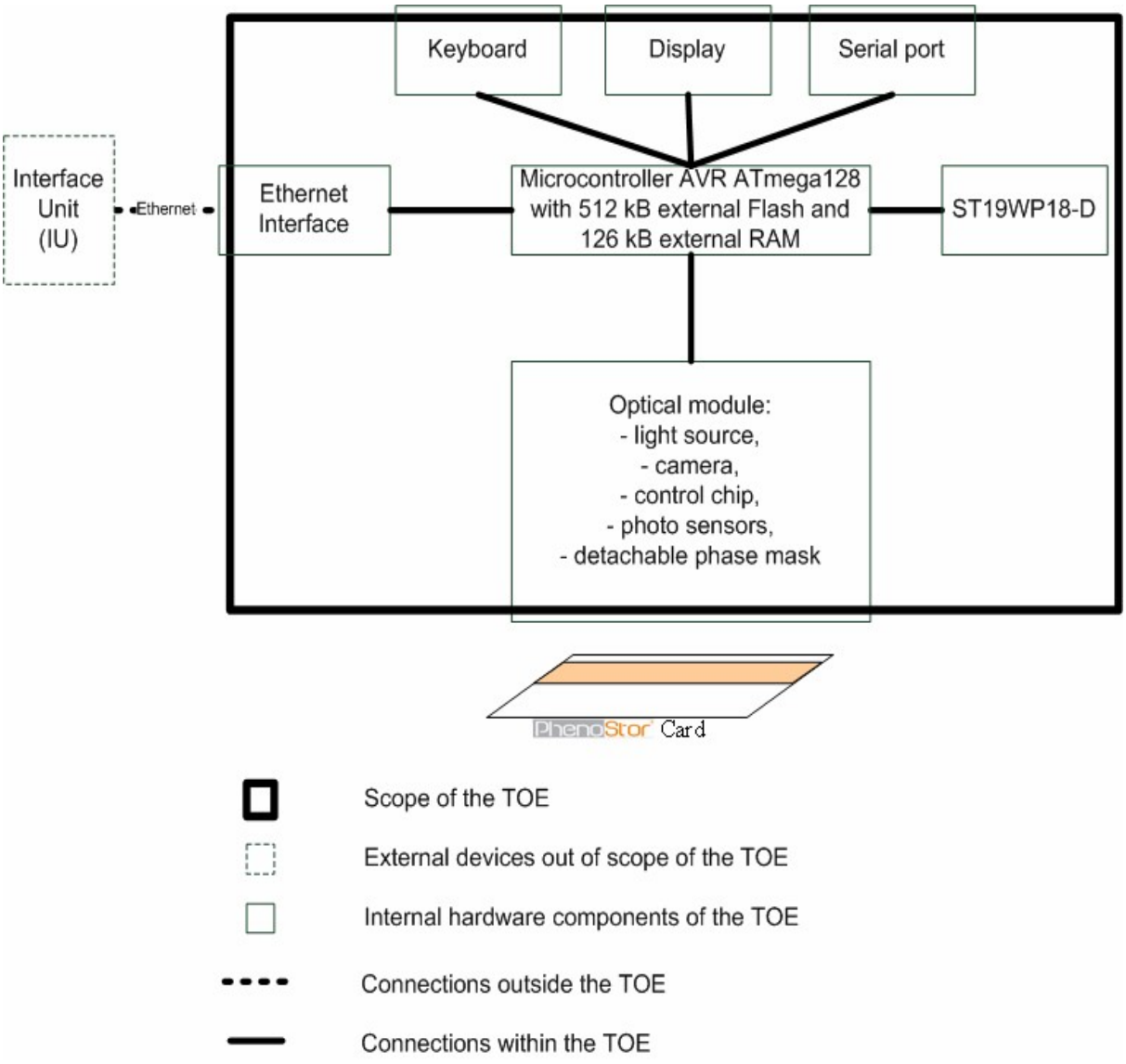


Figure 1: General hardware structure of the TOE and hardware external interfaces

The interface of a TOE component to the outside is symbolized by some part of the border of the component outside the Scope of the TOE (Figure 1). The core security component is the CC EAL5+ certified *8-bit security controller* `ST19WP18-D` by `STMicroelectronics` which is compliant with the Smartcard IC Platform Protection Profile BSI-PP-0002-2001 [6] and Protection Profile Smartcard Integrated Circuit PP/9806 [7]. The security controller takes over all cryptographic functions within the TOE and stores all key data of the TOE. The security controller, its certified dedicated software and the embedded software form the security relevant part of the TOE.

The *8-bit standard microcontroller* `ATmega128` with 128 KByte external RAM and 512 KByte external flash memory provides management capability and basic

communication functionality. Neither cryptographic nor other security relevant operations are performed here.

The standard microcontroller offers several low-level communication interfaces. First, there is an interface for receiving raw data (bit stream) from the optical module. Second, the standard microcontroller provides the communication interface (over the Ethernet interface) to and from the IU. Third, the standard microcontroller controls the low-level communication with the security controller. Finally, it handles peripheral devices (keyboard, display).

The *optical module* consists of a card slot, a light source, a camera, a control chip, photo sensors, and a detachable phase mask enabling analogue scrambling of the light beam. The phase mask can be used for analogue encryption [1] of holographically stored data. The analogue encryption scheme provides confidentiality and authenticity of the holographically stored data. The authenticity is given in terms of applying an integrity check after analogously decrypting the data. The analogue encryption can be applied on top of digital encryption and thereby enhances security. Furthermore, it is an effective protection method against cloning memory cards (copy protection), which is not provided by digital cryptography. Holographic data storage is an effective protection against cloning of memory cards per se. The analogue encryption enhances this protection method. However, analogue cryptography is not considered within the evaluation because there is lack of standard evaluation procedures to examine the level of security of analogue encryption methods.

The communication with the IU is implemented over the *Ethernet interface* which is a standard Ethernet controller with a connector. The *display* and *keyboard* are controlled by the IU through the standard microcontroller and are used for outputting graphical and text data and inputting information, respectively.

2.3 TOE software description

The software of the TOE consists of the following four modules (Figure 2):

- *SEC Module*: The functionality of this module covers all the security mechanisms needed for supporting the security features of the TOE. The SEC module consists in turn of two modules:
    - *SEC_ROM Module*: This module is physically placed in the ROM mask of the security controller ST19WP18-D. It contains some auxiliary routines for initializing the execution of SEC_EEPROM and accessing the EERPOM.
    - *SEC_EEPROM Module*: This module is physically placed in the EEPROM of the security controller ST19WP18-D. The SEC_EEPROM

code is written into the EEPROM within the personalization step (cf. 2.6). Its functionality covers all the security mechanisms needed for maintaining the security features of the TOE.

- *ESC Module*: The module runs on the standard microcontroller and is intended for the communication management between the SEC module, the COM module and external devices over the COM module.

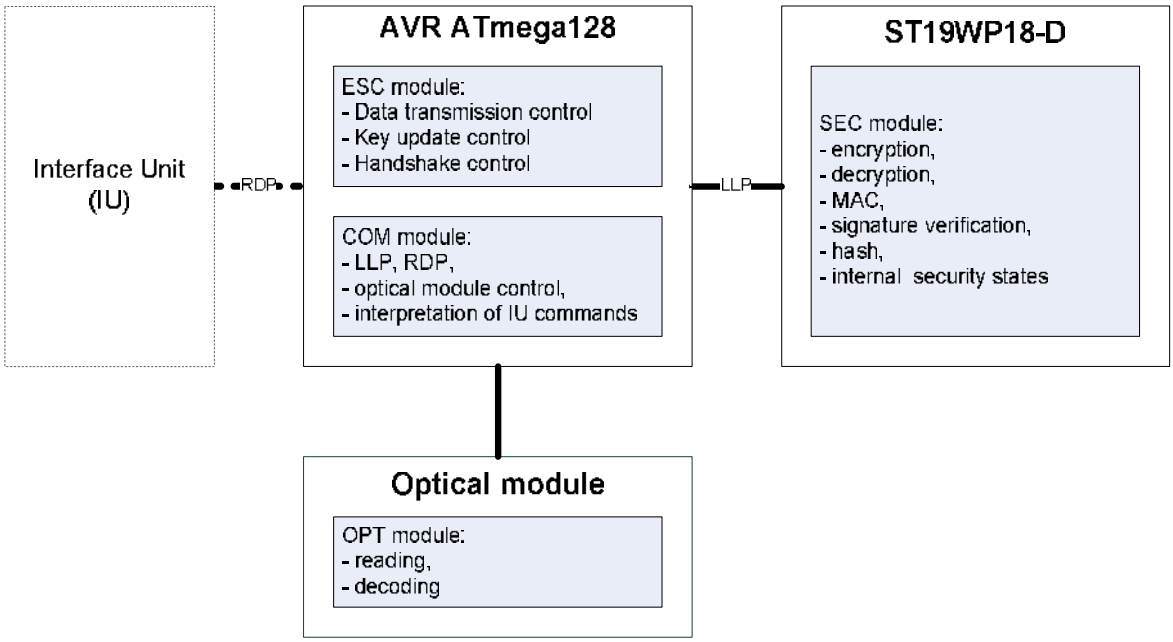


Figure 2: General software structure of the TOE

- *COM Module*: The module provides the overall control over the course of computations, manages the process of communication with external devices (including external security devices), and provides low-level external and internal interfaces on the side of the standard microprocessor.
- *OPT module*: The module controls the process of holographically reading out the data from a holographic memory card including the control over the optical components and low-level decoding the data in the optical module.

The *COM module* receives instructions from the IU and controls the communication between the components of the TOE. When needed the COM module uses the

security functionality of the SEC module on the security controller through the ESC module. The *ESC module* provides the following interfaces:

- *Handshake*: Before a trusted communication channel with the IU is set up, a handshake takes place in order to authenticate each other's identity and to negotiate cryptographical parameters of the trusted channel between the parties. The ESC module starts and controls the handshake procedure and uses the security functionality provided by the SEC module and IU. The communication with the SEC module and the IU takes place through the corresponding functionality of the COM module.
- *Key update*: The symmetric and asymmetric keys stored in the security controller can be updated in the field. The owner of the update key can initiate a mutual authentication procedure with the security controller and transmit the new keys to the security controller. This process is controlled by the ESC module. The communication with the SEC module and IU takes place through the corresponding functionality of the COM module.
- *Data transmission:* Once a trusted channel is established by the handshake procedure, protected data blocks from the holographic memory can be transmitted to the IU encrypted with a key known to the IU.

The ESC module makes use of the following low level communication interfaces provided by the COM module:

- Raw Data Protocol (RDP) for communication with external security devices over an Ethernet connection;
- Low Level Protocol (LLP) for communication with the security controller using a derivation of the LPT protocol (with four data paths instead of eight ones).

The *SEC module* implements all the security relevant functionalities of the TOE, including the support for personalization, key update, secure handshake with the IU, data decryption and signature verification, and secure data transmission to the IU. Additionally, the LLP for the communication with the standard microcontroller is implemented there.

The *OPT module* controls the process of optically reading the holographically stored data from a holographic card, its decoding and transmission to the standard controller.

2.4 External interfaces of TOE

The TOE provides the following external interfaces:

- Optical interface
    - Optical unit for readout of data from holographic memory card and transformation of optical signals to electrical signals by means of a camera.
- Display and key pad (human interfaces)
    - The card reader is equipped with a display and a key pad as interfaces to the human user. These interfaces are not security relevant, since no sensitive information is read / written from / to these devices.
- Electrical interfaces
    - Power supply

      This interface is needed for the power supply of the TOE.
    - Ethernet hardware interface for the IU

      This interface makes possible the communication of the TOE with the IU. It is based on the standard Ethernet protocol, a frame-based computer networking technology for local area networks.
    - Serial port interface

      This interface is connected to the AVR ATmega128. It is neither used nor does it provide any services.
- Software interfaces
    - Interfaces for the communication between the IU and the COM module for exchanging control information.
    - Interfaces for the communication between the IU and the SEC module over the ESC and COM modules.
    - Interfaces for the communication between the administrator’s device and the SEC module over the ESC and COM modules as well as between the SEC module and the IU over the ESC and COM modules.

2.5 TOE security features

The TOE provides a multi-layer security concerning the following aspects:

- Standard cryptographical methods are used to protect data of a holographic memory card on its way from the card to the IU, from the IU to the security controller and from the security controller to the IU.

- The data on the card are digitally encrypted and signed. Alternatively, the data may be digitally encrypted and MAC-protected. For the symmetric encryption the TripleDES algorithm with 2 keys is applied [8] (key length of 112 bit). The SHA1 function [8] is used for computing and checking digital signatures with RSA algorithm [8], [12]. The RSA modulo length can be selected at the card writing stage (1024-2048 bit). For the MAC-protection the TripleDES based MAC with 2 keys is used [8].
- The communication channel from the security controller to the IU is protected by the TripleDES algorithm with 2 keys [8] (key length of 112 bit) and the TripleDES based MAC [8]. The key establishment and authentication procedure uses a secure ISO/IEC 11770-2:1996 [9] standard key establishment protocol.
- All the keys within the TOE can be updated by the administrator using the ISO/IEC 11770-2:1996 [9] standard key establishment protocol (mechanism 6).
- The card reader system uses the certified cryptographic library of the CC EAL5+ certified ST19WP18-D controller which is protected against leakage attacks. Additional measures enhancing the side-channel resistance of the implementation are taken.
- The data on the holographic card can be analogue scrambled which is, however, not considered as security relevant in the course of the certification process. Nonetheless, the security feature clearly makes reading out the information from the holographic card more complicated.

2.6 TOE life cycle

The TOE life cycle is divided into four main stages (Figure 3), each consisting of one or several steps:

- *Development*: On this stage the software of SEC, ESC, COM, OPT modules and the TOE auxiliary hardware (except for the security controller) are either developed at `escrypt GmbH` (SEC, ESC) or delivered by `Bayer Innovation GmbH` (COM, OPT, TOE auxiliary hardware).
- *Testing*: This stage comprises two test steps. First, the local software and hardware components are tested. Second, the TOE software modules taken together are tested using all available simulators and hardware emulators.
- *Manufacturing*: This stage mainly consists of the following actions:

- issuing the ROM mask for the security controller (SEC_ROM module by escrypt GmbH) including the boot loader application to be placed into the executable EEPROM within the fixed pre-personalization,
- manufacturing the standard microcontroller and TOE auxiliary hardware and software (delivered by Bayer Innovation GmbH),
- ST19WP18-D production (STMicroelectronics), and
- TOE personalization including:
  - switching ST19WP18-D to the User mode,
  - writing the code of SEC_EEPROM module as well as production keys into the ST19WP18-D EEPROM in the User mode using T=0 like commands of the boot loader application,
  - resetting the ST19WP18-D in order to run the newly loaded SEC_EEPROM module,
  - generating keys and writing them into the ST19WP18-D EEPROM (escrypt GmbH) using a secure ISO/IEC 11770-2:1996 [9] standard key establishment protocol and the production keys written into the EEPROM via the boot loader application.

- *End-Usage*: Exploitation of the TOE by an end user.

The TOE life cycle and the life cycle of its components are topologically distributed over four sites:

- STMicroelectronics,
- escrypt GmbH,
- Bayer Innovation GmbH,
- End user.

The transitions step 4 – step 5 and step 5 – step 6 need trusted delivery with verification. Additionally, the TOE related keys must be delivered to the end user in a secure way (step 6 – step 7). All other delivery processes are either not security relevant or take place in the same secure production environment. Note that the TOE life cycle is compatible with that of ST19WP18-D as described in [8].

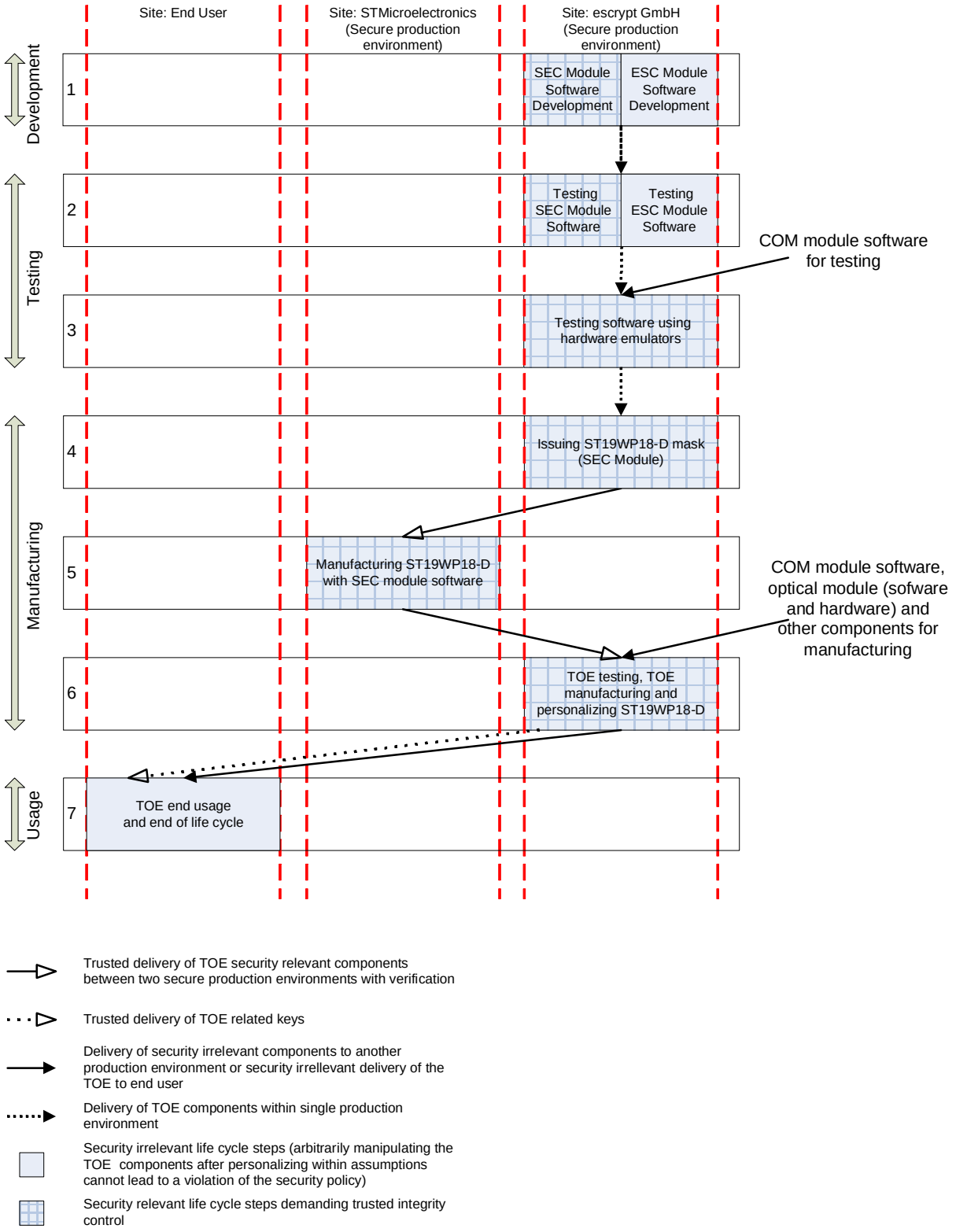


Figure 3: TOE life cycle.

2.7 TOE intended use

The TOE is designed for handling any kind of confidential and manipulation sensitive information, which is stored on a mobile storage medium and has to be processed digitally.

The administrator's unit, interface unit and card writer that are needed in the IT environment of the card reader for its functioning are provided by Bayer Innovation GmbH.

Data stored on the mobile storage medium is encrypted and signed (or MAC-protected) and, therefore, cannot be read or undetectably modified without knowing the corresponding keys. The corresponding keys are stored safely in the security controller, which is able to decrypt data and to verify signatures (and MACs), but which transfers decrypted data to the IU solely via trusted channels.

There is a variety of potential applications where the TOE can be applied. One potential application of the TOE is identification check for physical access control based on biometric verification of users. A user authenticates himself by a biometric characteristic (such as fingerprint, iris pattern, and so on), which is compared to reference data stored on the holographic memory card. Here, the biometric data stored on the card is signed/MAC-protected by a trust center. The signature/MAC is stored on the card, too. The biometric data is digitally encrypted. For identification, the card is read. The standard microcontroller forwards the data over the IU to the security controller which decrypts the data and verifies the signature/MAC. It then forwards the biometric data with some additional information (e.g. the claimed identity) to the IU, which in turn forwards it to external biometric devices in a secure way (e.g. over a TLS secured communication channel). There, the biometric template is compared with the life pattern to verify the identity. The biometric device returns the result via this secure channel to the IU which then takes appropriate action.

Another potential application of the TOE is the administration and transmission of medical data, e.g. a personal medical record. Such information is sensitive and should be protected against misuse. This can be ensured by the TOE: Medical information is encrypted and stored on a holographic memory card. The key for encryption can be derived from biometrics of the card owner, so that the owner is able to control the access to the stored information. A card reader is installed at a doctor´s surgery, together with a biometric authentication device. To give the doctor access to the information stored on the card, the card owner first has to put the card into the card reader. Then he has to present one of his biometric characteristics (fingerprint, iris pattern, voice, etc.) to the biometric authentication device. From the captured biometric data a key is derived. The key is used to decrypt the information, which is read from the card. Only if the right person presents the right biometric characteristics, the encrypted information can be read.

Once the basic principle of the TOE is understood, one can imagine a variety of further applications, where an authentication process (based on the data holographically stored on a card and supported by the operational environment) is

involved: financial transactions, ticket systems, border crossings, access to any kind of digital content, where DRM plays a role, logical access and so on.

3 TOE security environment

3.1 Assets

When a PhenoStor® card is written, the user can decide to mark certain data blocks as protected data. Before writing it on the card, the user must digitally sign (or MAC-protect) and encrypt each individual block of protected data. This data in the protected data block has to be protected against manipulation and disclosure within the TOE, i.e. the confidentiality, integrity and authenticity of the data has to be ensured. The necessary symmetric keys stored in the TOE have to be protected from manipulation and disclosure, i.e. the confidentiality, integrity and authenticity of the symmetric keys has to be ensured. The necessary public key stored in the TOE has to be protected from manipulation, i.e. the integrity and authenticity of the public key has to be ensured. The assets to be protected are denoted by:

- A.ProtectedData is the user data;
- A.SymkeyData is the symmetric key used for decrypting and MAC checking the data from a PhenoStor® card;
- A.PubkeySignature is the public key for checking the signature of the data from a PhenoStor® card;
- A.SymkeyIU is the symmetric key used for securing communication between the security controller and the IU;
- A.SymkeyRNG is used in the creation of the initial state of the deterministic random number generator;
- A.SymkeyAdmin is the symmetric key used by the administrator for updating A.SymkeyData, A.SymkeyIU, A.SymkeyAdmin, A.PubkeySignature.

All the assets and additionally the private key PrivkeySignature corresponding to A.PubkeySignature have to be protected against disclosure and manipulation in the TOE environment (see AE3 and AE5).

The notion *A.ProtectedData* is an abstract notion for the user information. This data is to be stored on a PhenoStor® card and should be protected from manipulation and disclosure. To be more precise, the notions of *encrypted A.ProtectedData* and *decrypted A.ProtectedData* are introduced.

Denote by *encrypted A.ProtectedData* a block of data which has been prepared for storage on a PhenoStor® card, i.e. the data has been

- marked as protected, encrypted and signed,

- marked as protected, MAC-protected and encrypted.

Encrypted A.ProtectedData therefore is the encrypted data whose integrity and authenticity can be checked.

The decrypted data whose integrity and authenticity has been checked is denoted by *decrypted A.ProtectedData*. The plain data before the encryption is also denoted by *decrypted A.ProtectedData*. The plaintext information within *decrypted A.ProtectedData* is A.ProtectedData.

Denote by “message” a string of bits. “To manipulate information” (e.g. A.ProtectedData or A.SymkeyData) shall mean to manipulate the message containing the information such that the message will be interpreted in a wrong way. “To operate on information” (e.g. A.ProtectedData or A.SymkeyData) shall mean to operate on the message containing the information.

Note that we distinguish between two types of authenticity which apply to different assets:

- Authenticity of data origin and time point (freshness) for A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin: It has to be guaranteed that not only the origin of these keys (administrator) is authentic, but also that the keys received by the TOE are up-to-date (e.g. protection from replay).
- Authenticity of data origin for a message containing A.ProtectedData, A.SymkeyData, A.PubkeySignature: It has to be guaranteed that the assets originate from the administrator. The time point is not critical.

In the following the identification of the corresponding authenticity type is unambiguously defined by the asset to which “authenticity” is applied.

The TOE does not guarantee the accessibility of any services nor data.

3.2 Assumptions

The following assumptions that cannot be directly enforced by the TOE are imposed on the TOE environment:

- AE1: The user marks as protected, digitally signs (or MAC-protects) and encrypts all the data that is to be stored on a PhenoStor® card and that he wants to be protected (i.e. the A.ProtectedData).
- AE2: All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, PrivkeySignature are generated according to the administration manual, that is, they are cryptographically good and strong.

- AE3: All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, `PrivkeySignature` in the environment (administrator, interface unit, card writer) are protected against manipulation and disclosure. The key A.PubkeySignature is protected against manipulation. The IU knows A.SymkeyIU, the security controller knows A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, the administrator knows all the keys (including `PrivkeySignature`).
- AE4: All random numbers in the environment (IU, administrator, card writer) used for cryptographic purposes are cryptographically good.
- AE5: A.ProtectedData is protected against disclosure and manipulation within the card writer, the IU and all other devices entitled to have access to any information about A.ProtectedData.
- AE6: The IU verifies that the TOE knows A.SymkeyIU before accepting data received from the TOE.
- AE7: The administrator's device verifies that the TOE knows the current A.SymkeyAdmin before sending data to the TOE.

3.3 Threats

- T1: An attacker could try to
  - manipulate A.ProtectedData on its way from the holographic memory card through the TOE to the IU or
  - manipulate A.ProtectedData on its way from the IU to the place in TOE where its processing starts,

  in order to provide the TOE with manipulated decrypted A.ProtectedData.
- T2: An attacker could try to reveal or manipulate A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin or to manipulate A.PubkeySignature by manipulating the program execution or the program code in the TOE (e.g. buffer overflow or glitches).
- T3: An attacker could try to get information about A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin through side-channel attacks (active and passive) on the TOE.
- T4: An attacker could try to manipulate A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature through active physical attacks on the TOE.

- T5: After its processing within the TOE, A.ProtectedData is about to be transmitted to the IU. An attacker could try to reveal or manipulate A.ProtectedData during this transmission as well as to generate manipulated A.ProtectedData and provide the IU with it on behalf of the TOE.
- T6: An attacker could try to exchange the IU against some other device (with some key known to him) in order to obtain A.ProtectedData.
- T7: An attacker could try to
    - o reveal or manipulate A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin or
    - o manipulate A.PubkeySignature

  on their way from the administrator to the place of their final storage within the TOE.
- T8: An attacker could try to exchange the administrator's device against some other device (with some key known to him) in order to manipulate (e.g. exchange) A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature in the TOE.
- T9: An attacker could try to reveal or manipulate A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin or to manipulate A.PubkeySignature by using the TOE personalization interface to manipulate program code or data.

3.4 Organizational security policies

The TOE reaches its specific security functionality only by a correct and effective usage of the underlying ST19WP18-D. In particular, this means that the SEC module and its developers must fulfill the assumptions as required in the ST19WP18-D Security Target [8], which refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7].

These relevant assumptions are suitably redefined in terms of organizational policies for the TOE as follows:

- P1: Procedures for the development, design, implementation and testing as required in the ST19WP18-D Security Target [8], which states an augmentation and refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], shall be applied.

- P2: Procedures for the manufacturing, delivery and storage as required in the ST19WP18-D Security Target [8], which refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], shall be applied.

Note that the two assumptions A.USE_DIAG and A.USE_SYS required for phase 7 in PP/9806 [7] are excluded here, since they are implemented by the TOE or restated as assumptions for the TOE environment (see also Section 8.1). Aug1.A.Key-Function from the ST19WP18-D Security Target [8] (see p.16) is as procedure covered by P1 but also directly addressed by the TOE (see also Section 8.1).

4 Security objectives

4.1 Security objectives for the TOE

- O1: The TOE must provide protection against
    - o manipulation of A.ProtectedData on its way from the holographic memory card through the TOE to the IU and
    - o manipulation of A.ProtectedData on its way from the IU to the place in TOE where its processing starts

  after receiving A.ProtectedData from the IU.
- O2: The TOE must provide protection against manipulation of the program execution or the program code in the TOE (e.g. buffer overflow or glitches) which (manipulation) could lead to disclosure or manipulation of A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin or to manipulation of A.PubkeySignature.
- O3: The TOE must provide protection against side-channel attacks (active and passive) on the TOE which could lead to the leakage of information about A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin.
- O4: The TOE must provide protection against active physical attacks on the TOE which could lead to manipulation of A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature.
- O5: The TOE must provide protection against disclosure and manipulation of A.ProtectedData during its transmission to the IU after processing within the TOE. This also means that it must be always possible to detect whether the data originates from the TOE.
- O6: The TOE must verify that the IU the TOE is connected to is entitled to receive A.ProtectedData.
- O7: The TOE must provide protection against
    - o disclosure and manipulation of A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and
    - o manipulation of A.PubkeySignature

  on their way from the administrator’s device to the place of their final storage within the TOE.

- O8: The TOE must verify that the administrator's device the TOE is connected to is entitled to update A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature in the TOE.
- O9: Only the TOE manufacturer is able to use the TOE personalization interface.

Additionally the following objectives are derived from the organizational policies for the TOE:

- OP1: The developer ensures that procedures for the development, design, implementation and testing as required in the ST19WP18-D Security Target [8], which states an augmentation and refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], are applied.
- OP2: The developer respectively the manufacturer ensures that procedures for the manufacturing, delivery and storage as required in the ST19WP18-D Security Target [8], which refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], are applied.

4.2 Security objectives for the environment

- OE1: The user must mark as protected, digitally sign (or MAC-protect) and encrypt all the data that is to be stored on a PhenoStor® card and that he wants to be protected (i.e. the decrypted A.ProtectedData).
- OE2: All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, PrivkeySignature must be generated according to the administration manual, that is, they must be cryptographically good and strong.
- OE3: All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, PrivkeySignature in the environment (administrator, interface unit, card writer) must be protected against manipulation and disclosure. The key A.PubkeySignature must be protected against manipulation. The IU knows A.SymkeyIU, the security controller knows A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, the administrator knows all the keys (including PrivkeySignature).
- OE4: All random numbers in the environment (IU, administrator, card writer) used for cryptographic purposes must be cryptographically good.

- OE5: A.ProtectedData must be protected against disclosure and manipulation within the card writer, the IU and all other devices entitled to have access to any information about A.ProtectedData.
- OE6: The IU must verify that the TOE knows A.SymkeyIU before accepting data received from the TOE.
- OE7: The administrator's device must verify that the TOE knows the current A.SymkeyAdmin before sending data to the TOE.

5 IT security requirements

5.1 TOE security functional requirements

This chapter defines the functional requirements for the TOE using functional components drawn from Common Criteria 2.2 Part 2 [3]. The minimum strength level for the TOE security functional requirements is `SOF-medium`. The strength of cryptographic algorithms is out of scope of the CC evaluation [5].

This chapter contains the descriptions of all derived functional requirements for the TOE.

A table which summarizes all functional requirements for the TOE and discusses their dependencies is given in Section 8.2.2.

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in paragraph 4.4.1.3.2 of Part 1 of the CC [2].

The **refinement** operation is used to add details to a component. Refinements that have been made by the ST authors are denoted as underlined text.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the ST authors are denoted as underlined text.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a cryptographic key. Assignments that have been made by the ST authors are denoted by showing as underlined text.

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash “/”, and the iteration indicator after the component identifier.

The Security Function Policy (SFP) **SFP_access_rules** as described below is used in the requirements “Complete access control (FDP_ACC.2)”, “Security attribute based access control (FDP_ACF.1)”, “Import of user data without security attributes (FDP_ITC.1.1)”, “Basic data exchange confidentiality (FDP_UCT.1/TOE)”, “Data

exchange integrity (FDP_UIT.1/Keys and FDP_UIT.1/Data)" and "Management of security attributes (FMT_MSA.1)".
The access control policy **SFP_access_rules** is only defined for the end usage phase of the TOE. Note, that access rules for personalisation are defined by management SFRs (FMT_MTD.1/Pers, see section 5.1.5), not by an explicit policy.

Denote by *A.ProtectedData decrypting and checking engine* a service provided by the TOE that reads in a block of encrypted A.ProtectedData, decrypts it and verifies the signature or MAC. It sends back a verification flag denoting the result of the authentication process as well as *decrypted A.ProtectedData*.

The role "administrator" is defined by the knowledge of A.SymkeyAdmin. This means that any subject that knows A.SymkeyAdmin is denoted by "administrator". In the same way the role "interface unit" is defined by the knowledge of A.SymkeyIU.

SFP_access_rules

The following subjects are covered by the policy:

card holder, interface unit, administrator, other person,
A.ProtectedData decrypting and checking engine

The following objects within the TOE are covered by the policy:

- data protection key A.SymkeyData,
- interface unit key A.SymkeyIU,
- random number generator initialization key A.SymkeyRNG,
- administrator access key A.SymkeyAdmin,
- user data signature key A.PubkeySignature,
- card identifier.

The following authentication methods are covered by the policy:

- trusted channel protocol (see SF.8).

The following security attributes for subjects are maintained by the TOE:

- interface unit key A.SymkeyIU,

- administrator access key A.SymkeyAdmin.

The following access methods are maintained by the TOE:

Access to the TOE's objects is allowed only

- by using the trusted channel protocol or
- by sending data directly to the TOE, which will then be interpreted:
  - encrypted A.ProtectedData to be decrypted and integrity checked by the A.ProtectedData decrypting and checking engine or
  - card identifier.

5.1.1 Class FCS: Cryptographic support

5.1.1.1 Cryptographic key management FCS_CKM

5.1.1.1.1 Cryptographic key generation FCS_CKM.1/TCP - Trusted channel protocol

FCS_CKM.1.1/TCP

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm trusted channel protocol and specified cryptographic key sizes 112 bit that meet the following: ISO/IEC 11770-2:1996(E) [9], 5.6 Key Establishment Mechanism 6.

Clarification: The trusted channel protocol (see SF.1), described in ISO/IEC 11770-2:1996(E) , mechanism 6, shall use good random numbers (see SF.8)

1. to generate nonces that are used in the trusted channel protocol for authentication and
2. to generate keying material to derive session keys (see SF.9).

5.1.1.1.2 Cryptographic key generation FCS_CKM.1/RND – Initialization of random number generator

FCS_CKM.1.1/RND

The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm usage of hardware unpredictable number

generator and specified cryptographic key sizes 176 bit that meet the following: initialization of DRNG E.4 as in AIS20 [10]**.**

Clarification: At start-up the (deterministic) random number generator must be initialized, i.e. its initial state (which can be regarded as a cryptographic key) has to be set. The 3DES key is constructed by XORing A.SymkeyRND with 112 bit of the SHA-1 hash value of a long enough output of the hardware unpredictable number generator of the ST19WP18-D described in the ST19WP18-D Security Target [8]. The first input to be encrypted with 3DES is 64 bit of the SHA-1 hash value of another long enough output of the hardware unpredictable number generator.

5.1.1.1.3 Cryptographic key destruction FCS_CKM.4/TCP - trusted channel protocol

FCS_CKM.4.1/TCP

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting keys with zeroes that meets the following: none.

Clarification: The TOE shall destroy the Triple-DES encryption and MAC session keys of the trusted channel protocol after termination of the trusted channel or reset (e.g. because of reaching a fail secure state according to FPT_FLS.1).

5.1.1.1.4 Cryptographic key destruction FCS_CKM.4/OUT – Outdated secure keys of TOE

FCS_CKM.4.1/OUT

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting keys with zeroes that meets the following: none.

Clarification: The TOE shall destroy the outdated keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature after a key update through the administrator. If keys are outdated and not yet entirely destroyed after a key update (e.g. because of reaching a fail secure state according to FPT_FLS.1), the TOE shall destroy the outdated keys during start-up.

5.1.1.1.5 Cryptographic key destruction FCS_CKM.4/RND – Secure key of random number generator

FCS_CKM.4.1/RND

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting with new key that meets the following: none.

Clarification: The TOE shall destroy the state of the deterministic random number generator (which can be regarded as a cryptographic key) during initial start-up, especially after reset (e.g. because of reaching a fail secure state according to FPT_FLS.1).

5.1.1.1.6 Cryptographic key destruction FCS_CKM.4/KDer – Secure keying material for key derivation

FCS_CKM.4.1/KDer

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting keys with zeroes that meets the following: none.

Clarification: The TOE shall destroy the keying material of the key derivation function (which can be regarded as a cryptographic key) after key derivation and after reset (e.g. because of reaching a fail secure state according to FPT_FLS.1), if it is not needed any more. Note that if the original A.SymkeyData, A.SymkeyIU, A.SymkeyAdmin and A.SymkeyRND stored in the TOE are keying material, it shall not be destroyed.

5.1.1.1.7 Cryptographic key destruction FCS_CKM.4/MAC3DES – Secure key for MAC3DES integrity check

FCS_CKM.4.1/MAC3DES

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting keys with zeroes that meets the following: none.

Clarification: The TOE shall destroy the key of the MAC3DES, after checking the MAC and after reset (e.g. because of reaching a fail secure state according to FPT_FLS.1).

5.1.1.2 Cryptographic operation FCS_COP

5.1.1.2.1 Cryptographic operation FCS_COP.1/SHA-1

FCS_COP.1.1/SHA-1

The TSF shall perform secure hash computation in accordance with a specified cryptographic algorithm SHA-1 and cryptographic key sizes none that meet the following: FIPS PUB 180-1 (chaining and padding) and ST19WP18-D Security Target [8], page 36.

Clarification: The secure hash computation is performed on data from enciphered A.ProtectedData for RSA digital signature verification. This requirement requires the secure hash computation to meet FIPS PUB 180-1 except in the parts of the computation that are performed by the ST19WP18-D's internal software which has been EAL 5+ evaluated.

The security controller ST19WP18-D implements a cryptographic operation "secure hash function" as stated in the ST19WP18-D Security Target [8]. The cryptographic algorithm in its FCS_COP.1/ST19WP18-D iteration is "revised Secure Hash Algorithm (SHA-1)" and the listed standard is "not applicable". In the ST19WP18-D's security function SF_AKCS_A is stated (see [8], page 46):

"In USER configuration, this security function implements the following standard hash function:

- SHA-1 hash function chaining blocks of 512 bits to get a 160 bits result.

In USER configuration, this security function provides to the SICESW developer very efficient primitives to design standard secure hash algorithms like SHA-1 or MD5."

5.1.1.2.2 Cryptographic operation FCS_COP.1/DataTripleDES

FCS_COP.1.1/DataTripleDES

The TSF shall perform decryption of encrypted A.ProtectedData in accordance with a specified cryptographic algorithm Triple-DES CBC Mode and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

Clarification: The intend of the requirement is to require to use in the implementation of this cryptographic operation the cryptographic operation implemented in the security controller ST19WP18-D which is explained in FCS_COP.1/ ST19WP18-D in the ST19WP18-D Security Target.

5.1.1.2.3 Cryptographic operation FCS_COP.1/TCPTripleDES

FCS_COP.1.1/TCPTripleDES

The TSF shall perform encryption and decryption within the trusted channel in accordance with a specified cryptographic algorithm Triple-DES CBC Mode and

cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

Clarification: The intend of the requirement is to require to use in the implementation of “encryption and decryption within the trusted channel” (see SF.1) the cryptographic operation implemented in the security controller ST19WP18-D which is explained in FCS_COP.1/ ST19WP18-D in the ST19WP18-D Security Target.

5.1.1.2.4 Cryptographic operation FCS_COP.1/RNDTripleDES

FCS_COP.1.1/RNDTripleDES

The TSF shall perform encryption with block cipher within generation of random numbers in accordance with a specified cryptographic algorithm Triple-DES and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

Clarification: The intend of the requirement is to require to use in the implementation of this cryptographic operation the cryptographic operation implemented in the security controller ST19WP18-D which is explained in FCS_COP.1/ ST19WP18-D in the ST19WP18-D Security Target.

5.1.1.2.5 Cryptographic operation FCS_COP.1/KDF

FCS_COP.1.1/KDF

The TSF shall perform encryption with block cipher within key derivation in accordance with a specified cryptographic algorithm Triple-DES and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

Clarification: The intend of the requirement is to require to use in the implementation of this cryptographic operation the cryptographic operation implemented in the security controller ST19WP18-D which is explained in FCS_COP.1/ ST19WP18-D in the ST19WP18-D Security Target.

5.1.1.2.6 Cryptographic operation FCS_COP.1/RSASignature

FCS_COP.1.1/RSASignature

The TSF shall perform verification of the RSA digital signature in accordance with a specified cryptographic algorithm RSA signature verification and cryptographic key

sizes between 1024 bit and 2048 bit that meet the following: RSA signature verification PKCS #1 v2.1 [12] in combination with FCS_COP.1/SHA-1**.**

Clarification: The verification of the RSA digital signature is performed on the RSA digital signature of the SHA-1 hash of enciphered A.ProtectedData. The intention of the requirement is to require to use in the implementation of this cryptographic operation the cryptographic operation *RSA recovery* implemented in the security controller ST19WP18-D which is explained in FCS_COP.1/ST19WP18-D in the ST19WP18-D Security Target [8], page 36. The padding is required to be as defined in PKCS #1 v2.1 [12].

5.1.1.2.7 Cryptographic operation FCS_COP.1/MAC3DES

FCS_COP.1.1/MAC3DES

The TSF shall perform MAC operation in accordance with a specified cryptographic algorithm Triple-DES CBC Mode and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

Clarification: The MAC verification is performed on decrypted A.ProtectedData. The intention of the requirement is to require to use in the implementation of this cryptographic operation the cryptographic operation implemented in the security controller ST19WP18-D which is explained in FCS_COP.1/ST19WP18-D in the ST19WP18-D Security Target.

5.1.1.3 Generation of random numbers FCS_RND

5.1.1.3.1 Quality metric for random numbers FCS_RND.1

FCS_RND.1.1

The TSF shall provide a mechanism to generate random numbers that meet resistance against an attacker with moderate attack potential in the sense of AIS20 [10]**.**

5.1.2 Class FDP: User Data Protection

5.1.2.1 Access control policy (FDP_ACC)

The following SFRs require the TOE to enforce the security policy **SFP_access_rules**. Note that all subjects, objects, security attributes, access methods and access rules are defined already in this policy.

5.1.2.1.1.1 *Complete access control*

The **SFP_access_rules** determine the protection of A.ProtectedData and the protection of the cryptographic keys stored in the card reader.

FDP_ACC.2.1

The TSF shall enforce the **SFP_access_rules** on the subjects card holder, interface unit, administrator, other person, A.ProtectedData decrypting and checking engine and the objects data protection key A.SymkeyData, interface unit key A.SymkeyIU, random number generator initialization key A.SymkeyRNG, administrator access key A.SymkeyAdmin, user data signature key A.PubkeySignature, A.ProtectedData, card identifier and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2

The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

5.1.2.2 Access control functions (FDP_ACF)

5.1.2.2.1 FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1

The TSF shall enforce the **SFP_access_rules** to objects based on the following: all subjects and objects together with their respective security attributes as defined in **SFP_access_rules**.

FDP_ACF.1.2

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

Rule_3: The administrator can update the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature in the TOE.

Rule_4: All subjects can send data to the A.ProtectedData decrypting and checking engine that are interpreted as encrypted A.ProtectedData or as card identifier.

Rule_6: The interface unit can receive data from the A.ProtectedData decrypting and checking engine.

Rule_8: The A.ProtectedData decrypting and checking engine has read access to the key A.SymkeyData.

FDP_ACF.1.3

The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: none.

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the following rules:

Rule_1: No subject has read access to any of the keys A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin in the TOE.

Rule_2: No subject but the administrator can update the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature in the TOE.

Rule_5: No subject but the interface unit can receive data from the A.ProtectedData decrypting and checking engine.

Rule_7: No subject but the A.ProtectedData decrypting and checking engine has read access to the key A.SymkeyData.

5.1.2.3 Export to outside TSF control (FDP_ETC)

5.1.2.3.1 FDP_ETC.2 Export of user data with security attributes

FDP_ETC.2.1

The TSF shall enforce the access control policy **SFP_access_rules** when exporting user data, controlled under the SFP(s), outside of the TSC.

FDP_ETC.2.2

The TSF shall export the user data with the user data's associated security attributes.

FDP_ETC.2.3

The TSF shall ensure that the security attributes, when exported outside the TSC, are unambiguously associated with the exported user data.

FDP_ETC.2.4

The TSF shall enforce the following rules when user data is exported from the TSC:

- The verification error flag is sent out after verification (see **FDP_ITC.1.3**).

5.1.2.4 Import from outside TSF control (FDP_ITC)

5.1.2.4.1 FDP_ITC.1 Import of user data without security attributes

FDP_ITC.1.1
The TSF shall enforce the access control policy **SFP_access_rules** when importing user data, controlled under the SFP, from outside of the TSC.

FDP_ITC.1.2
The TSF shall ignore any security attributes associated with the user data when imported from outside the TSC.

FDP_ITC.1.3
The TSF shall enforce the following rules when importing user data controlled under the SFP from outside the TSC:

- There is neither integrity nor authenticity verification performed for the imported data interpreted as card identifier.
- If the integrity or authenticity verification of other imported data fails, the data is disregarded. A verification error flag is set for export.
- Imported data exceeding the expected length is truncated. In this case an error message is transmitted by the TOE to the sender.

5.1.2.5 Residual information protection (FDP_RIP)

5.1.2.5.1 FDP_RIP.1 Subset residual information protection

FDP_RIP.1.1
The TSF shall ensure that any previous information content of a resource is made unavailable upon the deallocation of the resource from the following objects: A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, derived keys stored in RAM.

5.1.2.6 Stored data integrity (FDP_SDI)

5.1.2.6.1 FDP_SDI.2 Stored data integrity monitoring and action

FDP_SDI.2.1

The TSF shall monitor user data stored within the TSC for integrity errors of TSF program code and personalization data that could lead to a violation of the SFP on all objects, based on the following attributes: checksum.

FDP_SDI.2.2

Upon detection of a data integrity error, the TSF shall interrupt the processing of the data and, if possible, reset the operation.

5.1.2.7 Inter-TSF user data confidentiality transfer protection (FDP_UCT)

5.1.2.7.1 FDP_UCT.1/TOE

5.1.2.7.2 Basic data exchange confidentiality

FDP_UCT.1.1/TOE

The TSF shall enforce the **SFP_access_rules** to be able to transmit and receive objects in a manner protected from unauthorised disclosure.

5.1.2.8 Inter-TSF user data integrity transfer protection (FDP_UIT)

5.1.2.8.1 Data exchange integrity FDP_UIT.1/Keys

FDP_UIT.1.1/Keys

The TSF shall enforce the **SFP_access_rules** to be able to transmit and receive user data in a manner protected from modification, deletion, insertion, and replay errors.

FDP_UIT.1.2/Keys

The TSF shall be able to determine on receipt of user data, whether modification, deletion, insertion and replay has occurred.

Clarification: FDP_UIT.1/Keys is about the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature that the administrator wants to replace with new keys.

5.1.2.8.2 Data exchange integrity FDP_UIT.1/Data

FDP_UIT.1.1/Data

The TSF shall enforce the **SFP_access_rules** to be able to transmit and receive user data in a manner protected from modification errors.

FDP_UIT.1.2/Data

The TSF shall be able to determine on receipt of user data, whether modification has occurred.

Clarification: FDP_UIT.1/Data is about the user data A.ProtectedData that the interface unit wants to have decrypted and integrity checked.

5.1.3 Class FIA: Identification and authentication

5.1.3.1 User authentication (FIA_UAU)

5.1.3.1.1 FIA_UAU.1 Timing of authentication

FIA_UAU.1.1

The TSF shall allow sending data to the A.ProtectedData decrypting and checking engine on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2

The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

Clarification: Possible users are the interface unit (requesting access to the A.ProtectedData decrypting and checking engine) and the administrator's device (requesting to overwrite cryptographic keys in the security controller).

5.1.3.2 User identification (FIA_UID)

5.1.3.2.1 FIA_UID.1 Timing of identification

FIA_UID.1.1

The TSF shall allow sending data to the A.ProtectedData decrypting and checking engine on behalf of the user to be performed before the user is identified.

FIA_UID.1.2

The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

Clarification: Possible users are the interface unit (requesting access to the decrypted A.ProtectedData) and the administrator's device (requesting to overwrite cryptographic keys in the security controller).

5.1.4 Class FTP: Trusted path/channels

5.1.4.1 Inter-TSF trusted channel (FTP_ITC)

5.1.4.1.1 FTP_ITC.1/TOE Inter-TSF trusted channel

FTP_ITC.1.1/TOE

The TSF shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

Clarification: The notion “protection from modification” shall include that the authenticity of the data sent through the trusted channel is verified, including its point of time. This means that no replay attacks shall be possible.

FTP_ITC.1.2/TOE

The TSF shall permit the TSF to initiate communication via a trusted channel.

FTP_ITC.1.3/TOE

The TSF shall initiate communication via a trusted channel for:

1. sending decrypted A.ProtectedData from the TOE to the interface unit,
2. storing a set of new keys in the TOE.

Clarification: Whenever the interface unit wants to use the A.ProtectedData decrypting and checking engine, it sends encrypted A.ProtectedData to the TOE which causes the A.ProtectedData decrypting and checking engine to decrypt and integrity check A.ProtectedData and the TOE to initiate a trusted channel to the interface unit. Whenever the administrator’s device wants to store a set of new keys in the reader, it notifies the TOE which causes it to initiate a trusted channel to the administrator’s device.

Security Management

5.1.5 Class FMT: Security management

5.1.5.1 Specification of Management Functions (FMT_SMF)

5.1.5.1.1 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1

The TSF shall be capable of performing the following security management functions:

1. Initialization
2. Personalization
3. Modification of the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature.

Clarification: Initialization is performed at every startup of the TOE. Personalization is performed by the TOE manufacturer during the assembly of the TOE before delivery to the end user. Modification of the keys can be performed by an administrator during the end usage phase of the TOE.

5.1.5.2 Security management roles (FMT_SMR)

5.1.5.2.1 Security roles (FMT_SMR.1)

FMT_SMR.1.1

The TSF shall maintain the roles interface unit, administrator, TOE manufacturer.

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

Clarification: The TOE authenticates interface unit and administrator by the trusted channel protocol using the keys A.SymkeyIU and A.SymkeyAdmin. The TOE manufacturer is authenticated by using the TOE personalization interface (ST19WP18-D in the Issue mode protected by a transport key) that he closes by switching to the User mode (in which the loading of the embedded software is performed) and therefore before delivery of the TOE to the end user.

5.1.5.3 Management of security attributes (FMT_MSA)

5.1.5.3.1 Management of security attributes (FMT_MSA.1)

FMT_MSA.1.1

The TSF shall enforce the access control policy **SFP_access_rules** to restrict the ability to modify, the security attributes A.SymkeyIU, and A.SymkeyAdmin to the administrator.

Clarification: This requirement is for the end usage phase only. Before the end usage phase the TOE manufacturer writes the corresponding keys to the TOE during personalization.

5.1.5.3.2 Secure security attributes (FMT_MSA.2)

FMT_MSA.2.1

The TSF shall ensure that only secure values are accepted for security attributes.

Clarification: This requirement addresses the secure nonce, keying material and derived session key in the trusted channel protocol.

5.1.5.4 Management of TSF data (FMT_MTD)

5.1.5.4.1 Management of TSF data (FMT_MTD.1/Pers) – Personalization

FMT_MTD.1.1/Pers

The TSF shall restrict the ability to write the personalization data to the TOE manufacturer.

Clarification: This requirement addresses only the personalization phase before the end usage phase. During personalization the manufacturer writes a.o. the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature. These can be changed by the administrator during the end usage phase. The administrator cannot write any other personalization data (e.g. ST19WP18-D EEPROM program code).

5.1.5.4.2 Management of TSF data (FMT_MTD.1/Keys) – Key management

FMT_MTD.1.1/Keys

The TSF shall restrict the ability to write the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature to the administrator.

Clarification: This requirement is for the end usage phase only. Before the end usage phase the TOE manufacturer writes the corresponding keys to the TOE during personalization.

General Security Functions

The TOE shall prevent inherent and forced illicit information flow for some data specified below. The security functional requirement FPT_EMSEC.1 addresses the inherent leakage. With respect to forced leakage they have to be considered in combination with the security functional requirements “Failure with preservation of secure state (FPT_FLS.1)” and “TSF testing (FPT_TST.1)” on the one hand and

“Resistance to physical attack (FPT_PHP.3)” on the other. The SFR “TSF domain separation (FPT_SEP.1)” prevents manipulation of the security features and misuse of TOE functions.

5.1.6 Class FPT: Protection of the TSF

5.1.6.1 TOE Emanation (FPT_EMSEC)

5.1.6.1.1 TOE Emanation (FPT_EMSEC.1)

FPT_EMSEC.1.1

The TOE shall not emit observable physical phenomena (e.g. variations in the power consumption, timing of signals, electromagnetic radiation) in excess of what an attacker with moderate attack potential can observe enabling access to

1. session key and random numbers (including the state of the deterministic random number generator) used in the trusted channel,

and

2. A.ProtectedData,
3. A.SymkeyData,
4. A.SymkeyIU,
5. A.SymkeyRNG,
6. A.SymkeyAdmin,
7. A.PubkeySignature.

FPT_EMSEC.1.2

The TSF shall ensure all the users are unable to use the following interface TOE internal electrical contacts accessable by an attacker with moderate attack potential to gain access to

1. session key and random numbers (including the state of the deterministic random number generator) used in the trusted channel,

and

2. A.ProtectedData,
3. A.SymkeyData,
4. A.SymkeyIU,
5. A.SymkeyRNG,
6. A.SymkeyAdmin,

7. A.PubkeySignature.

5.1.6.2 Fail secure (FPT_FLS)

5.1.6.2.1 FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1

The TSF shall preserve a secure state when the following types of failures occur:

1. exposure to operating conditions where therefore a malfunction could occur,
2. failure detected by TSF according to FPT_TST.1.

5.1.6.3 TSF physical protection (FPT_PHP)

5.1.6.3.1 FPT_PHP.3 Resistance to physical attack

FPT_PHP.3.1

The TSF shall resist physical manipulation and physical probing to the TSF by responding automatically such that the TSP is not violated.

5.1.6.4 TSF self test (FPT_TST)

5.1.6.4.1 FPT_TST.1 TSF testing

FPT_TST.1.1

The TSF shall run a suite of self tests during initial start-up to demonstrate the correct operation of the TSF.

FPT_TST.1.2

The TSF shall provide authorised users with the capability to verify the integrity of TSF data.

FPT_TST.1.3

The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.

Clarification: The TOE runs a set of self tests at the request of the authorised user (manufacturer) and a set of self tests automatically to detect failure and to preserve of secure state according to FPT_FLS.1 during the operational use. These tests are described in the ST19WP18-D Security Target [8].

5.1.6.5 Domain separation (FPT_SEP)

5.1.6.5.1 FPT_SEP.1 TSF domain separation

FPT_SEP.1.1

The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2

The TSF shall enforce separation between the security domains of subjects in the TSC.

Clarification: The security domain in the TOE is implemented by the security controller ST19WP18-D.

5.1.7 Extended components definition

This security target uses components defined as extensions to CC part 2 [3]. The family FCS_RND is justified because cryptographically good random numbers are needed for the Inter-TSF trusted channel (FTP_ITC/TOE). The family FPT_EMSEC is justified because it is needed to formulate requirements about side channel resistance. The definitions of these components (FCS_RND and FPT_EMSEC) are taken from the Protection Profile for the Electronic Health Card [11]. They are reproduced in the following.

5.1.7.1 Definition of the Family FCS_RND

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

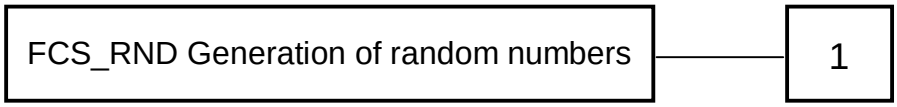
The family “Generation of random numbers (FCS_RND)” is specified as follows.

FCS_RND Generation of random numbers

Family behaviour

> This family defines quality requirements for the generation of random numbers which are intended to be use for cryptographic purposes.

Component levelling:



FCS_RND.1 Generation of random numbers requires that random numbers meet a defined quality metric.

Management: FCS_RND.1

There are no management activities foreseen.

Audit: FCS_RND.1

There are no actions defined to be auditable.

FCS_RND.1 Quality metric for random numbers

Hierarchical to: No other components.

FCS_RND.1.1 The TSF shall provide a mechanism to generate random numbers that meet [assignment: *a defined quality metric*].

Dependencies: No dependencies.

5.1.7.2 Definition of the Family FPT_EMSEC

The family “TOE Emanation (FPT_EMSEC)” is specified as follows.

Family behaviour

This family defines requirements to mitigate intelligible emanations.

Component levelling:

FPT_EMSEC.1 TOE emanation has two constituents:

FPT_EMSEC.1.1 Limit of Emissions requires to not emit intelligible emissions enabling access to TSF data or user data.

FPT_EMSEC.1.2 Interface Emanation requires not emit interface emanation enabling access to TSF data or user data.

Management: FPT_EMSEC.1

There are no management activities foreseen.

Audit: FPT_EMSEC.1

There are no actions defined to be auditable.

FPT_EMSEC.1 TOE Emanation

Hierarchical to: No other components.

FPT_EMSEC.1.1 The TOE shall not emit [assignment: *types of emissions*] in excess of [assignment: *specified limits*] enabling access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].

FPT_EMSEC.1.2 The TSF shall ensure [assignment: *type of users*] are unable to use the following interface [assignment: *type of connection*] to gain access to [assignment: *list of types of TSF data*] and [assignment: *list of types of user data*].

Dependencies: No other components.

5.2 TOE security assurance requirements

The TOE shall meet the set of the security assurance requirements encompassing the evaluation assurance level EAL3 (methodically tested and checked) according to Common Criteria 2.2 Part 3 [4]. These assurance components are listed in the following Table 1.

Assurance class	Assurance component ID	Assurance components
Configuration management	ACM_CAP.3	Authorization controls
	ACM_SCP.1	TOE configuration management coverage
Delivery and operation	ADO_DEL.1	Delivery
	ADO_IGS.1	Installation, generation and start-up
Development	ADV_FSP.1	Functional specification
	ADV_HLD.2	High-level design
	ADV_RCR.1	Representation correspondence
Guidance documents	AGD_ADM.1	Administrator guidance
	AGD_USR.1	User guidance
Life cycle support	ALC_DVS.1	Development security
Tests	ATE_COV.2	Coverage
	ATE_DPT.1	Depth
	ATE_FUN.1	Functional tests
	ATE_IND.2	Independent tests
Vulnerability assessment	AVA_MSU.1	Misuse
	AVA_SOF.1	Strength of TOE security functions
	AVA_VLA.1	Developer vulnerability analysis

Table 1: TOE Security assurance requirements (EAL3)

5.3 Security requirements for the TOE environment

Here the security requirements for the IT environment of the TOE as well as the ones of the non-IT environment of the TOE are defined.

5.3.1 Security requirements for the non-IT environment of the TOE

The security requirements for the non-IT environment of the TOE follow directly from the security objectives for the environment defined in Section 4.2 and are the following:

- SRE1: The user shall mark as protected, digitally sign (or MAC-protect) and encrypt all the data that is to be stored on a PhenoStor® card and that he wants to be protected (i.e. the decrypted A.ProtectedData).
- SRE2: All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, PrivkeySignature shall be generated according to the administration manual, that is, they shall be cryptographically good and strong.
- SRE3: All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, PrivkeySignature in the environment (administrator, interface unit, card writer) shall be protected against manipulation and disclosure. The key A.PubkeySignature shall be protected against manipulation. The IU shall know A.SymkeyIU, the security controller shall know A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, and the administrator shall know all the keys (including PrivkeySignature).
- SRE4: All random numbers in the environment (IU, administrator, card writer) used for cryptographic purposes shall be cryptographically good.
- SRE5: A.ProtectedData shall be protected against disclosure and manipulation within the card writer, the IU and all other devices entitled to have access to any information about A.ProtectedData.
- SRE6: The IU shall verify that the TOE knows A.SymkeyIU before accepting data received from the TOE.
- SRE7: The administrator’s device shall verify that the TOE knows the current A.SymkeyAdmin before sending data to the TOE.

5.3.2 Security requirements for the TOE IT environment

Here the essential functional requirements for the IT environment of the TOE are defined using functional components drawn from Common Criteria 2.2 Part 2 [3].

This chapter contains the descriptions of all derived essential functional requirements for the IT environment of the TOE.

The CC allows several operations to be performed on functional requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in Paragraph 4.4.1.3.2 of Part 1 of the CC [2].

The **refinement** operation is used to add details to a component. Refinements that have been made by the ST authors are denoted as underlined text.

The **selection** operation is used to select one or more options provided by the CC in stating a requirement. Selections that have been made by the ST authors are denoted as underlined text.

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a cryptographic key. Assignments that have been made by the ST authors are denoted by showing as underlined text.

The **iteration** operation is used when a component is repeated with varying operations. Iteration is denoted by showing a slash “/”, and the iteration indicator after the component identifier.

In the IT environment of the TOE there are three major components whose security and correct operation have impact on the TOE (see also Section 1.2):

- AU (administrator unit),
- IU (interface unit),
- CW (card writer).

The functional requirements for the IT environment of the TOE are represented according to these three devices in the IT environment of the TOE.

In the definition of the security functional requirements for the TOE IT environment the Security Function Policy (SFP) **SFP_access_rules** as described in Section 5.1 is used.

5.3.3 Class FCS: Cryptographic support

5.3.3.1 Cryptographic key management FCS_CKM

5.3.3.1.1 Cryptographic key generation FCS_CKM.1/AU_TCP - Trusted channel protocol for AU

FCS_CKM.1.1/AU_TCP

The AU shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm trusted channel protocol and specified cryptographic key sizes 112 bit that meet the following: ISO/IEC 11770-2:1996(E) [9], 5.6 Key Establishment Mechanism 6.

5.3.3.1.2 Cryptographic key generation FCS_CKM.1/IU_TCP - Trusted channel protocol for IU

FCS_CKM.1.1/IU_TCP

The IU shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm trusted channel protocol and specified cryptographic key sizes 112 bit that meet the following: ISO/IEC 11770-2:1996(E) [9], 5.6 Key Establishment Mechanism 6.

5.3.3.1.3 Cryptographic key destruction FCS_CKM.4/AU_TCP - trusted channel protocol for AU

FCS_CKM.4.1/AU_TCP

The AU shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting keys that meets the following: none.

5.3.3.1.4 Cryptographic key destruction FCS_CKM.4/IU_TCP - trusted channel protocol for IU

FCS_CKM.4.1/IU_TCP

The IU shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method overwriting keys that meets the following: none.

5.3.3.2 Cryptographic operation FCS_COP

5.3.3.2.1 Cryptographic operation FCS_COP.1/CW_SHA-1 – Card Writer

FCS_COP.1.1/CW_SHA-1

The CW shall perform secure hash computation in accordance with a specified cryptographic algorithm SHA-1 and cryptographic key sizes none that meet the following: FIPS PUB 180-1 (chaining and padding) and ST19WP18-D Security Target [8], page 36.

Clarification: The CW should be able to generate RSA signatures for the blocks of data that are to be stored as encrypted A.ProtectedData on the holographic memory cards. For this purpose the SHA-1 hash function is needed (see also Section 5.1.1.2, where the corresponding functional requirement for the TOE is derived).

5.3.3.2.2 Cryptographic operation FCS_COP.1/CW_DataTripleDES

FCS_COP.1.1/CW_DataTripleDES

The CW shall perform encryption of decrypted A.ProtectedData in accordance with a specified cryptographic algorithm Triple-DES CBC Mode and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

Clarification: The intend of the requirement is to require of the CW to use an implementation of this cryptographic operation compatible with that specified in FCS_COP.1/ ST19WP18-D in the ST19WP18-D Security Target. This operation is necessary for the CW to generate encrypted A.ProtectedData that will be written to holographic memory cards.

5.3.3.2.3 Cryptographic operation FCS_COP.1/AU_TCPTripleDES

FCS_COP.1.1/AU_TCPTripleDES

The AU shall perform encryption and decryption within the trusted channel in accordance with a specified cryptographic algorithm Triple-DES CBC Mode and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

5.3.3.2.4 Cryptographic operation FCS_COP.1/IU_TCPTripleDES

FCS_COP.1.1/IU_TCPTripleDES

The IU shall perform encryption and decryption within the trusted channel in accordance with a specified cryptographic algorithm Triple-DES CBC Mode and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

5.3.3.2.5 Cryptographic operation FCS_COP.1/CW_RSASignature

FCS_COP.1.1/CW_RSASignature

The CW shall perform generation of the RSA digital signature in accordance with a specified cryptographic algorithm RSA signature generation and cryptographic key sizes between 1024 bit and 2048 bit that meet the following: RSA signature verification PKCS #1 v2.1 [12] in combination with FCS_COP.1.1/CW_SHA-1**.**

Clarification: The CW should be able to generate RSA signatures for the blocks of data that are to be stored as encrypted A.ProtectedData on the holographic memory cards.

5.3.3.2.6 Cryptographic operation FCS_COP.1/AU_MAC3DES

FCS_COP.1.1/AU_MAC3DES

The AU shall perform MAC operation in accordance with a specified cryptographic algorithm Triple-DES CBC Mode and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

5.3.3.2.7 Cryptographic operation FCS_COP.1/IU_MAC3DES

FCS_COP.1.1/IU_MAC3DES

The IU shall perform MAC operation in accordance with a specified cryptographic algorithm Triple-DES CBC Mode and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

5.3.3.2.8 Cryptographic operation FCS_COP.1/CW_MAC3DES

FCS_COP.1.1/CW_MAC3DES

The CW shall perform MAC operation in accordance with a specified cryptographic algorithm Triple-DES CBC Mode and cryptographic key size 112 bit that meet the following: see ST19WP18-D Security Target [8], page 36.

Clarification: The CW should be able to generate MAC values for the blocks of data that are to be stored as encrypted A.ProtectedData on the holographic memory cards.

5.3.4 Class FDP: User Data Protection

5.3.4.1 Inter-TSF user data confidentiality transfer protection (FDP_UCT)

5.3.4.1.1 Basic data exchange confidentiality FDP_UCT.1/AU

FDP_UCT.1.1/AU

The AU shall enforce the **SFP_access_rules** to be able to transmit and receive objects in a manner protected from unauthorised disclosure.

5.3.4.1.2 Basic data exchange confidentiality FDP_UCT.1/IU

FDP_UCT.1.1/IU

The IU shall enforce the **SFP_access_rules** to be able to transmit and receive objects in a manner protected from unauthorised disclosure.

5.3.4.2 Inter-TSF user data integrity transfer protection (FDP_UIT)

5.3.4.2.1 Data exchange integrity FDP_UIT.1/AU_Keys

FDP_UIT.1.1/AU_Keys

The AU shall enforce the **SFP_access_rules** to be able to transmit and receive user data in a manner protected from modification, deletion, insertion, and replay errors.

FDP_UIT.1.2/AU_Keys

The AU shall be able to determine on receipt of user data, whether modification, deletion, insertion and replay has occurred.

5.3.4.2.2 Data exchange integrity FDP_UIT.1/IU_Data

FDP_UIT.1.1/IU_Data

The IU shall enforce the **SFP_access_rules** to be able to transmit and receive user data in a manner protected from modification errors.

FDP_UIT.1.2/IU_Data

The IU shall be able to determine on receipt of user data, whether modification has occurred.

5.3.4.3 Import from outside TSF control (FDP_ITC)

5.3.4.3.1 Import of user data with security attributes FDP_ITC.2

FDP_ITC.2.1

The IU shall enforce the **SFP_access_rules** when importing user data, controlled under the IU security policy, from outside of the IU.

FDP_ITC.2.2

The IU shall use the security attributes associated with the imported user data.

FDP_ITC.2.3

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

FDP_ITC.2.4

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

FDP_ITC.2.5

The IU shall enforce the following rules when importing user data controlled under the IU security policy from outside the IU:

- the IU shall react in an appropriate way corresponding to the end user security policy, if the integrity flag received together with decrypted A.ProtectedData is not valid. In such a case the decrypted A.ProtectedData must not be treated as integer and valid.

5.3.5 Class FTP: Trusted path/channels

5.3.5.1 Inter-TSF trusted channel (FTP_ITC)

5.3.5.1.1 FTP_ITC.1/AU Inter-TSF trusted channel

FTP_ITC.1.1/AU

The AU shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

Clarification: The notion “protection from modification” shall include that the authenticity of the data sent through the trusted channel is verified, including its point of time. This means that no replay attacks shall be possible.

FTP_ITC.1.2/AU

The AU shall permit the remote trusted IT product to initiate communication via a trusted channel.

FTP_ITC.1.3/AU

The AU shall initiate communication via a trusted channel for:

- updating a set of keys in the TOE.

5.3.5.1.2 FTP_ITC.1/IU Inter-TSF trusted channel

FTP_ITC.1.1/IU

The IU shall provide a communication channel between itself and a remote trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

Clarification: The notion “protection from modification” shall include that the authenticity of the data sent through the trusted channel is verified, including its point of time. This means that no replay attacks shall be possible.

FTP_ITC.1.2/IU

The IU shall permit the remote trusted IT product to initiate communication via a trusted channel.

FTP_ITC.1.3/IU

The IU shall initiate communication via a trusted channel for:

- receiving decrypted A.ProtectedData from the TOE.

6 TOE summary specification

6.1 TOE security functions

SF.1 Trusted channel with confidentiality, integrity and authenticity

All security relevant data exchange with the TOE (except reading a PhenoStor® card and transferring its data unchanged to the interface unit and sending encrypted A.ProtectedData from the IU to the TOE) is done through a trusted channel which is established by the trusted channel protocol (ISO/IEC 11770-2:1996(E), see [9]). The building of the trusted channel is started by the TOE, generating nonces and keying material using the deterministic random number generator in SF.8. Keys are derived from the keying material by using the key derivation function in SF.9. The data transport through the trusted channel is done by padding a data block, appending a MAC, using one derived key, and encrypting this longer data block with 3DES using another derived key. This MAC-protected and encrypted data block is sent through the tunnel. If an answer is to be sent back, two more derived keys are used to MAC-protect and encrypt the data block that is sent back through the tunnel. At the end the tunnel is closed, i.e. the tunnel is not used any more and the nonces and all derived keys are overwritten with zeroes. The destruction of the keying material is described in the Key derivation function SF.9.

Whenever the administrator's device communicates with the TOE or data is to be sent to the interface unit, the trusted channel protocol authenticates the device to the TOE and vice versa and builds a trusted channel which guarantees the confidentiality, integrity and authenticity of all data sent through this channel, protecting the trusted channel from replay attacks. All security relevant functionality is performed in the Security domain described in SF.10, in such a way that the trusted channel logically ends in the Security domain (not just anywhere within the TOE).

SF.2 TOE state integrity protection during and after interruption of the normal procedure

Whenever the normal procedure of the TOE is interrupted by a power failure or any other fatal error, the TOE is interrupted and the embedded software is restarted (starting with the start-up function). The only security relevant state of the TOE which is preserved through a restart is the cryptographic keys stored in the Security domain described in SF.10. The protection of the keys is done by using transactions when writing the keys. Whenever the administrator wishes to store a set of new keys in the TOE, he sends the keys to the TOE.

If the TOE is interrupted during these transactions, the TOE remains in a secure state, because keys are only used, if they are completely stored and ready to be used. The start-up functions destroys remaining keys.

At the start-up of the Security domain (described in SF.10), automatic self tests of the security controller are performed, that test the functionality and data integrity of the security controller as described in the ST19WP18-D Security Target [8].

This TOE state integrity protection (SF.2 and SF.10) guarantees that no corrupted user data or keys are used after a fatal error.

SF.3 Decryption and verification of encrypted A.ProtectedData

Whenever the interface unit wants to decrypt encrypted A.ProtectedData (which presumably comes from the PhenoStor® card that has just been inserted in the card reader), the interface unit sends this data to the TOE. The SEC module in the Security domain (see SF.10) decrypts the data with 3DES using a key derived from A.SymkeyData and the card identifier. Then it checks its integrity and authenticity either by comparing a MAC3DES with a second key derived from A.SymkeyData or by an RSA signature verification (PKCS #1 v2.1 [12] in combination with SHA-1) using A.PubkeySignature. Then the derived keys are destroyed. The destruction of the input of the derivation function is described in SF.9.

The verification flag is set according to the MAC3DES respectively the signature verification. Then the SEC module in the Security domain initiates a trusted channel from the Security domain to the interface unit (see SF.1). At this point the interface unit has authenticated itself to the security controller, this is part of the initiation of the trusted channel. The interface unit is entitled to get A.ProtectedData in plain text. The TOE then sends the verification flag and decrypted A.ProtectedData to the interface unit.

SF.4 Protection against unauthorized personalization

In the end usage phase of the TOE, this function protects the personalization interface from being used by preventing the execution of the boot loader application. The boot loader application in the EEPROM is completely overwritten by the embedded software during the personalization. Any change of the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature requires the knowledge of A.SymkeyAdmin, and unauthorized manipulation is prevented. Together, this protects the TOE from unauthorized personalization.

SF.5 Access denial

This function denies every access controlled by the **SFP_access_rules** during the end usage phase that is not explicitly granted by SF.6 and SF.7. It also denies every attempt to use the personalization interface during the end usage phase.

SF.6 Protection against unauthorized change of keys

The role “administrator” is defined by the knowledge of A.SymkeyAdmin stored in the TOE. No other subject than the administrator may overwrite cryptographic keys in the security controller. The TOE’s security controller establishes a trusted channel with confidentiality, integrity and authenticity (see SF.1) to the administrator’s device, verifying its authenticity using the key A.SymkeyAdmin. The Security domain (see SF.10) then receives the set of new keys from the administrator’s device through the trusted channel and closes the trusted channel. The storing of the keys using transactions and the destruction of the old keys is described in SF.2.

SF.7 Protection against unauthorized access to decrypted A.ProtectedData

Whenever the interface unit wants to decrypt encrypted A.ProtectedData (which presumably comes from the PhenoStor® card that has just been inserted in the card reader), the interface needs to use the A.ProtectedData decrypting and checking engine.

The interface unit sends encrypted A.ProtectedData to the TOE. The data is decrypted and its integrity and authenticity is verified using SF.3.

The SEC module in the Security domain sets a verification flag according to the result of the authentication/integrity check. It then initiates a trusted channel from the Security domain to the interface unit (see SF.1). At this point the interface unit has authenticated itself to the security controller, this is part of the initiation of the trusted channel. The interface unit is entitled to get A.ProtectedData in plain text. The TOE then sends the verification flag and decrypted A.ProtectedData to the interface unit and closes the trusted channel. No other subject than the interface unit is entitled to receive the decrypted A.ProtectedData.

SF.8 Generation of random numbers

The random numbers provided by this security function are the output of a deterministic random number generator as described in E.4 on page 17 of the

Application Notes and Interpretation of the Scheme (AIS 20), see [10]. It uses the block cipher 3DES with 112 bit for updating its state.

At start-up the (deterministic) random number generator is initialized, i.e. its initial state has to be set. In particular this is done at every reset.

The 3DES key is constructed by XORing A.SymkeyRND with 112 bit of the SHA-1 hash value of a long enough output of the hardware unpredictable number generator of the ST19WP18-D described in the ST19WP18-D Security Target [8]. The first input to be encrypted with 3DES is 64 bit of the SHA-1 hash value of another long enough output of the hardware unpredictable number generator.

SF.9 Key derivation functions (KDFs)

The key derivation functions for SF.1 and SF.3 use Triple-DES (see [8]) as their main component:

- For SF.1 two types of KDFs are needed:
  - One computing a session key from the negotiated parameters. This key is derived from two nonces by performing the bitwise exclusive OR operation over them (according to ISO/IEC 11770-2:1996(E), see [9]). To make this operation side-channel resistant Triple-DES in ECB and OFB (see [8]) modes with A.SymkeyRND is used.
  - Another one computing encryption and MAC-protection keys from A.SymkeyAdmin, A.SymkeyIU and the negotiated session key. This function takes A.SymkeyAdmin, A.SymkeyIU or the session key as 3DES key and some constants as IVs (encryption and MAC-protection keys are derived using different values of IV). Then it generates the necessary number of key-stream blocks using 3DES in OFB mode with the inputs described above.
- For SF.3 a KDF producing two card keys from A.SymkeyData is needed: one key for the decryption of encrypted A.ProtectedData and another one for the verification of its MAC-value. This keys are derived from A.SymkeyData, the imported card identifier and a constant (different constants for decryption and MAC-verification keys) using 3DES in OFB mode (see [8]) with A.SymkeyData as 3DES key. The corresponding constant and card identifier form the IV. Then the KDF generates the needed number of outputs to correspondingly obtain decryption and MAC-verification keys.

After usage, the keying material is destroyed by overwriting it with zeroes, if it is not used any more. This explicitly means that keying material in the RAM of the security

controller (that implements the Security domain described in SF.10) is overwritten with zeroes, whereas keying material in the EEPROM of the security controller is left unchanged. Keying material that is left unchanged can be: A.SymkeyData, A.SymkeyIU or A.SymkeyAdmin.

SF.10 Security domain

The TSF in the TOE run in a special Security domain (the security controller ST19WP18-D) that protects the Security domain from

- interference and tampering by untrusted subjects,
- availability of any information about A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin,
- unauthorized access to
    - o A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature,
    - o session key or random numbers (including the state of the deterministic random number generator) used in the trusted channel by the administrator and by the interface unit

  by using observable physical phenomena or TOE internal electrical contacts.

The Security domain maintains a secure state

- during exposure to operating conditions where therefore a malfunction could occur and
- if a failure is detected by TSF according to FPT_TST.1 (start-up testing described in SF.2).

It is resistant to physical manipulation and physical probing, responding automatically such that the TSP is not violated. It monitors all TSF program code and personalization data within the Secure domain for integrity errors that could lead to a violation of the SFP. Upon detection of a data integrity error, the Security domain interrupts the processing of the data and, if possible, resets the operation. It runs a suite of self tests during initial start-up to demonstrate the correct operation of the TSF.

The level of the strength of the TOE's security functions is claimed as SOF-medium.

The TOE security functions which are realized by probabilistic or permutational mechanisms are the following:

- SF8: The generation of random numbers (E.4 from AIS20 [10], recursive call of 3DES with 2 keys) using SHA1 hashed ST19WP18-D hardware unpredictable numbers and an additional secret key A.SymkeyRND.
- SF1: Data and subject authentication during the initialization and usage of the trusted channel. For the authentication the following mechanisms are used:
  - Message Authentication Code (MAC) is the 3DES CBC encryption algorithm with two keys and a constant IV. The last 64-bit ciphertext block is used as the resulting MAC value. See also [8].
  - 128-bit challenges used to guarantee unpredictability of input and freshness of established keys according to [9]. The generator of random numbers (see SF8) is used to produce the challenges on the side of the TOE.

  Note that these mechanisms of the security function SF1 are also used by security functions SF6 and SF7.
- SF3: Verification of the integrity and authenticity of encrypted A.ProtectedData. The following mechanisms are used:
  - Message Authentication Code (MAC) is the 3DES CBC encryption algorithm with two keys and a constant IV. The last 64-bit ciphertext block is used as the resulting MAC value. See also [8].
  - The SHA-1 hash function with 160-bit outputs (described in the ST19WP18-D Security Target [8]) is used in the RSA-PSS signature method to bind the encrypted A.ProtectedData to the signature.

Although SF10 contains a further (partially) probabilistic mechanism used for EEPROM integrity protection in the ST19WP18-D security controller; this Security Target does not claim any SOF for it. This is due to the fact that the mechanism cannot be directly attacked. The corresponding attack path consists of two main steps: The first one is physical (invasive) and is infeasible for an attacker with the assumed low attack potential. The second step is of probabilistic nature, but it is impossible without the first one.

Note that the TOE´s cryptographic algorithms itself may also be analysed with permutational or probabilistic methods but this is not in the scope of the CC evaluation [5].

6.2 Assurance measures

Appropriate assurance measures are employed by the producer of the TOE to satisfy the security assurance requirements defined in Section 5.2. The TOE fulfills the evaluation assurance level EAL3. The present document (Security Target for the PhenoStor® Card Reader GRE100010) serves as evidence of satisfying the requirements of class ASE. In addition to the TOE the following documents are delivered within the evaluation process:

- Configuration management documentation (according to ACM_CAP.3 and ACM_SCP.1);
- Delivery and operation documentation (according to ADO_DEL.1, ADO_IGS.1);
- Development documentation (according to ADV_FSP.1, ADV_HLD.2 and ADV_RCR.1);
- Handbooks (according to AGD_ADM.1 and AGD_USR.1);
- Life cycle documentation (according to ALC_DVS.1);
- Test documentation (according to ATE_COV.2, ATE_DPT.1, ATE_FUN.1 and ATE_IND.2);
- Vulnerability assessment documentation (according to AVA_MSU.1, AVA_SOF.1 and AVA_VLA.1).

As stated in 1.2 the security of the TOE is based upon the STMicroelectronics smartcard integrated circuit ST19WP18-D with its dedicated software which is EAL5+ certified in compliance with Common Criteria 2.2 and in accordance with the Protection Profiles [6] and [7]. Therefore also the following two documents from the evaluation process of ST19WP18-D are provided within the evaluation process:

- ST19WP18-D Security Target (ST-Lite) [8];
- Evaluation Report Lite (ETR-Lite) for ST19WP18-D.

7 PP claims

No conformation to any protection profile is provided.

8 Rationale

8.1 Security objectives rationale

This security objectives rationale demonstrates that the stated security objectives are traceable to the threats, assumptions and organizational policies (OSPs) described in Chapter 3 and that the security objectives are suitable to cover them. The following table shows, which security objectives for the TOE and for the TOE environment addresses which threat, correspond to which assumption and cover which OSP.

	O1	O2	03	04	O5	O6	O7	O8	O9	OP1	OP2	OE1	OE2	OE3	OE4	OE5	OE6	OE7
T1	X																	
T2		X																
T3			X															
T4				X														
T5					X													
T6						X												
T7							X							X				X
T8								X										
T9									X									
P1			X							X								
P2											X							
AE1												X						
AE2													X					
AE3														X				
AE4															X			
AE5																X		
AE6																	X	
AE7																		X

Table 2: Mapping of security objectives to threats, assumptions and OSPs

The table shows, that for every threat, assumption or OSP there is at least one security objective and vice versa.

The following explanations describe for every threat, assumption and OSP, how they are covered by the security objectives.

Assuming the fulfillment of the assumptions on the TOE environment AE1-AE7 the threats T1-T9 are countered by the objectives for the TOE O1-O9 as follows:

- **T1** “An attacker could try to
  - manipulate A.ProtectedData on its way from the holographic memory card through the TOE to the IU or
  - manipulate A.ProtectedData on its way from the IU to the place in TOE where its processing starts,

  in order to provide the TOE with manipulated decrypted A.ProtectedData.”

  is directly countered by the objective

  O1 “The TOE must provide protection against
  - manipulation of A.ProtectedData on its way from the holographic memory card through the TOE to the IU and
  - manipulation of A.ProtectedData on its way from the IU to the place in TOE where its processing starts

  after receiving A.ProtectedData from the IU.”

- **T2** “An attacker could try to reveal or manipulate A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin or to manipulate A.PubkeySignature by manipulating the program execution or the program code in the TOE (e.g. buffer overflow or glitches).”

  is directly countered by the objective

  O2 “ The TOE must provide protection against manipulation of the program execution or the program code in the TOE (e.g. buffer overflow or glitches) which (manipulation) could lead to disclosure or manipulation of A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin or to manipulation of A.PubkeySignature.”

- **T3** “An attacker could try to get information about A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin through side-channel attacks (active and passive) on the TOE.”

  is directly countered by the objective

O3 “The TOE must provide protection against side-channel attacks (active and passive) on the TOE which could lead to the leakage of information about A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin.”

- **T4** ”An attacker could try to manipulate A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature through active physical attacks on the TOE.”

  is directly countered by the objective

  O4 “ The TOE must provide protection against active physical attacks on the TOE which could lead to manipulation of A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature.”

- **T5** ”After its processing within the TOE, A.ProtectedData is about to be transmitted to the IU. An attacker could try to reveal or manipulate A.ProtectedData during this transmission as well as to generate manipulated A.ProtectedData and provide the IU with it on behalf of the TOE.”

  is directly countered by the objective

  O5 “ The TOE must provide protection against disclosure and manipulation of A.ProtectedData during its transmission to the IU after processing within the TOE. This also means that it must be always possible to detect whether the data originates from the TOE.”

- **T6** ”An attacker could try to exchange the IU against some other device (with some key known to him) in order to obtain A.ProtectedData.”

  is directly countered by the objective

  O6 “ The TOE must verify that the IU the TOE is connected to is entitled to receive A.ProtectedData.”

- **T7** “An attacker could try to

  - reveal or manipulate A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin or
  - manipulate A.PubkeySignature

  on their way from the administrator to the place of their final storage within the TOE.”

  is countered by the objective

  O7 “The TOE must provide protection against

- o disclosure and manipulation of A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and
- o manipulation of A.PubkeySignature

on their way from the administrator to the place of their final storage within the TOE."

in conjunction with OE 3 and OE 7, stating:

OE3 "All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, `PrivkeySignature` in the environment (administrator, interface unit, card writer) must be protected against manipulation and disclosure. The key A.PubkeySignature must be protected against manipulation. The IU knows A.SymkeyIU, the security controller knows A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, the administrator knows all the keys (including `PrivkeySignature`)."

OE7 "The administrators's device must verify that the TOE knows the current A.SymkeyAdmin before sending data to the TOE."

- **T8** "An attacker could try to exchange the administrator's device against some other device (with some key known to him) in order to manipulate (e.g. exchange) A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature in the TOE."

  is directly countered by the objective

  O8 "The TOE must verify that the administrator's device the TOE is connected to is entitled to update A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature in the TOE."

- **T9** "An attacker could try to reveal or manipulate A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin or to manipulate A.PubkeySignature by using the TOE personalization interface to manipulate program code or data."

  countered by the objective

  O9 "Only the TOE manufacturer is able to use the TOE personalization interface."

  since the attacker cannot use the TOE personalization interface.

The policies P1 and P2 are covered by the security objectives as follows.

Note first that the SEC module and its developers must fulfill the assumptions as required in the ST19WP18-D Security Target [8], which refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7].

These assumptions are reformulated as policies P1 and P2 for the TOE excluding the two assumptions A.USE_DIAG and A.USE_SYS required for phase 7 in PP/9806 [7], since they are directly implemented by the TOE (A.USE_DIAG, finally covered by O5, O6, O7, O8, since these objectives cover the required secure communication protocols and procedures used between smartcard and terminal) respectively are partly implemented by the TOE and are partly restated as assumptions on the TOE environment (A.USE_SYS, finally supported by the assumptions AE1-AE7 outside the TOE and by O1-O9 within the TOE, since all of them contribute to the integrity and confidentiality of sensitive data stored/handled by the system).

Then

- **P1** “Procedures for the development, design, implementation and testing as required in the ST19WP18-D Security Target [8], which states an augmentation and refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], shall be applied.”

  is covered by the objective

  OP1 “The developer ensures that procedures for the development, design, implementation and testing as required in the ST19WP18-D Security Target [8], which states an augmentation and refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], are applied.”

  and supportively by the objective for the TOE **O3,** since **P1** also covers the assumption Aug1.Key-Function (see ST19WP18-D Security Target [8], p. 16) which addresses the usage of key-dependent functions, which shall be implemented in the smartcard embedded software in a way that they are not susceptible to leakage attacks, and **O3** directly covers this.

- **P2** “ Procedures for the manufacturing, delivery and storage as required in the ST19WP18-D Security Target [8], which refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], shall be applied.”

  is directly covered by the objective

  OP2 “ The developer respectively the manufacturer ensures that procedures for the manufacturing, delivery and storage as required in the ST19WP18-D Security Target [8], which refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], are applied.”

The assumptions on the TOE environment are completely covered by the objectives for the TOE environment as follows:

- **AE1** “The user marks as protected, digitally signs (or MAC-protects) and encrypts all the data that is to be stored on a PhenoStor® card and that he wants to be protected (i.e. the decrypted A.ProtectedData).” is directly implemented by the objective **OE1** “The user must mark as protected, digitally sign (or MAC-protect) and encrypt all the data that is to be stored on a PhenoStor® card and that he wants to be protected (i.e. the decrypted A.ProtectedData).”
- **AE2** “All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, PrivkeySignature are generated according to the administration manual, that is, they are cryptographically good and strong.” is directly implemented by the objective **OE2** “All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, PrivkeySignature must be generated according to the administration manual, that is, they must be cryptographically good and strong.”
- **AE3** “All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, PrivkeySignature in the environment (administrator, interface unit, card writer) are protected against manipulation and disclosure. The key A.PubkeySignature is protected against manipulation. The IU knows A.SymkeyIU, the security controller knows A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, the administrator knows all the keys (including PrivkeySignature).” is directly implemented by the objective **OE3** “All the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, PrivkeySignature in the environment (administrator, interface unit, card writer) must be protected against manipulation and disclosure. The key A.PubkeySignature must be protected against manipulation. The IU knows A.SymkeyIU, the security controller knows A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, the administrator knows all the keys (including PrivkeySignature).”
- **AE4** “All random numbers in the environment (IU, administrator, card writer) used for cryptographic purposes are cryptographically good.” is directly implemented by the objective **OE4** “All random numbers in the environment (IU, administrator, card writer) used for cryptographic purposes must be cryptographically good.”

- **AE5** “A.ProtectedData is protected against disclosure and manipulation within the card writer, the IU and all other devices entitled to have access to any information about A.ProtectedData.” is directly implemented by the objective **OE5** “A.ProtectedData must be protected against disclosure and manipulation within the card writer, the IU and all other devices entitled to have access to any information about A.ProtectedData.”
- **AE6** “The IU verifies that the TOE knows A.SymkeyIU before accepting data received from the TOE.” is directly implemented by the objective **OE6** “The IU must verify that the TOE knows A.SymkeyIU before accepting data received from the TOE.”
- **AE7** “The administrator’s device verifies that the TOE knows A.SymkeyAdmin before sending data to the TOE.” is directly implemented by the objective **OE7** “The administrators’s device must verify that the TOE knows A.SymkeyAdmin before sending data to the TOE.”

8.2 Security requirements rationale

This section demonstrates that the set and combination of the security requirements (SFRs and SARs) defined in Section 5.1 are suitable to satisfy the identified security objectives for the TOE. It shows that each of the functional security requirements corresponds to at least one of the security objectives for the TOE and vice versa (excluding two objectives derived from OSPs, which are covered by SARs). It also demonstrates that all dependencies between the identified SFRs are satisfied except for several cases which are explained separately. Additionally, the mapping between the security objectives for the environment and the security requirements for the IT as well as the non-IT security requirements is demonstrated. The section also includes the security assurance requirements rationale, the rationale on mutual support and internal consistency for the security requirements and the strength of function rationale.

8.2.1 Security objectives – security requirements

8.2.1.1 Security functional requirements coverage

The following table shows, which SFRs for the TOE support which security objectives of the TOE. The table shows, that every objective is supported by at least one SFR and that every SFR supports at least one objective.

Security Requirements vs. Security Functions	O.1	O.2	O.3	O.4	O.5	O.6	O.7	O.8	O.9
FCS_CKM.1/TCP					X	X	X	X	
FCS_CKM.1/RND					X	X	X	X	
FCS_CKM.4/TCP					X		X		
FCS_CKM.4/OUT			X		X	X	X	X	
FCS_CKM.4/RND					X	X	X	X	
FCS_CKM.4/KDer	X				X		X		
FCS_CKM.4/MAC3DES	X				X		X		
FCS_COP.1/SHA-1	X								
FCS_COP.1/DataTripleDES	X								
FCS_COP.1/TCPTripleDES					X		X		
FCS_COP.1/RNDTripleDES					X	X	X	X	
FCS_COP.1/KDF	X				X		X		
FCS_COP.1/RSASignature	X								
FCS_COP.1/MAC3DES	X				X		X		
FCS_RND.1					X	X	X	X	
FDP_ACC.2						X		X	
FDP_ACF.1						X		X	
FDP_ETC.2					X				
FDP_ITC.1							X		
FDP_RIP.1			X			X		X	
FDP_SDI.2		X		X					
FDP_UCT.1/TOE			X		X		X		
FDP_UIT.1/Keys				X			X		
FDP_UIT.1/Data				X	X				

FIA_UAU.1						X		X	
FIA_UID.1						X		X	
FTP_ITC.1/TOE					X	X	X	X	
FMT_SMF.1								X	X
FMT_SMR.1						X		X	X
FMT_MSA.1						X		X	
FMT_MSA.2					X	X	X	X	
FMT_MTD.1/Pers		X							X
FMT_MTD.1/Keys								X	
FPT_EMSEC.1			X						
FPT_FLS.1		X		X					X
FPT_PHP.3		X	X	X					
FPT_TST.1		X							X
FPT_SEP.1		X	X	X					X

Table 3: Security requirements vs. security objectives

8.2.1.2 Functional requirements sufficiency

The security objective **O.1** is reached by calculating either a MAC3DES or by verifying an RSA signature. The security objective is implemented by the following SFRs:

- the SFR FCS_COP.1/KDF requires to use Triple-DES in the key derivation function to derive a 3DES key from A.SymkeyData necessary for decryption of encrypted A.ProtectedData, and (if integrity and authenticity is verified using a MAC3DES) to derive a MAC3DES key,
- the SFR FCS_CKM.4/KDer requires to destroy any copied key material used for key derivation, such that it cannot be used by an attacker,

- the SFR FCS_COP.1/DataTripleDES requires the decryption of encrypted A.ProtectedData which is necessary to detect manipulation,
- the SFR FCS_COP.1/MAC3DES requires to calculate the MAC3DES (if integrity and authenticity is verified using a MAC3DES),
- the SFR FCS_COP.1/SHA-1 requires to calculate the SHA-1 hash of enciphered A.ProtectedData. The hash value is needed for signature verification (if integrity and authenticity is verified using a RSA signature),
- the SFR FCS_COP.1/RSASignature requires to verify the RSA signature of the SHA-1 hash (if integrity and authenticity is verified using a RSA signature),
- the SFR FCS_CKM.4/MAC3DES requires to destroy the MAC3DES key (if integrity and authenticity is verified using a MAC3DES) after calculating the MAC3DES, such that it cannot be used by an attacker.

The security objective **O.2** is implemented by the following SFRs:

- the SFR FDP_SDI.2 requires to prevent integrity errors of the TSF program code and personalization data that could lead to a violation of the SFP,
- the SFR FMT_MTD.1/Pers requires to restrict the ability to exchange program code to the TOE manufacturer,
- the SFR FPT_FLS.1 requires the preservation of a secure state in view of protection of the assets,
- the SFR FPT_PHP.3 requires resistance against physical attack in view of protection of the assets,
- the SFRs FPT_TST.1 requires appropriate testing of parts of the TOE in view of protection of the assets,
- the SFR FPT_SEP.1 requires to separate the domain for execution of the TSF, such that this domain can be specially protected without interference from other parts of the TOE.

The security objective **O.3** is implemented by the following SFRs:

- the SFRs FDP_RIP.1 and FCS_CKM.4/OUT require residual information about the outdated keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin is made unavailable respectively destroyed, such that it cannot leak to the outside,

- the SFR FDP_UCT.1/TOE requires to protect A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG and A.SymkeyAdmin (and parts thereof) from unauthorised disclosure,
- the SFR FPT_EMSEC.1 requires to protect A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG and A.SymkeyAdmin from access through observable physical phenomena,
- the SFR FPT_PHP.3 requires to resist against physical attack that could disclose A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG or A.SymkeyAdmin (or parts thereof),
- the SFR FPT_SEP.1 requires to separate the domain for execution of the TSF, such that this domain can be specially protected without interference from other parts of the TOE.

The security objective **O.4** is implemented by the following SFRs:

- the SFR FDP_SDI.2 requires to protect decrypted A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature against integrity errors while stored,

- the SFR FDP_UIT.1/Keys and FDP_UIT.1/Data requires to protect decrypted A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature against integrity errors while transmitted or received,

- the SFR FPT_FLS.1 requires to preserve a secure state in view of protection of the assets during an active physical attack,

- the SFR FPT_PHP.3 requires to resist against physical attack that could lead to manipulation of A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG or A.SymkeyAdmin,

- the SFR FPT_SEP.1 requires to separate the domain for execution of the TSF, such that this domain can be specially protected against active physical attacks.

The security objective **O.5** is implemented by the following SFRs:

- the SFR FCS_CKM.1/TCP requires to generate a secure session key for the data transmission,

- the SFR FCS_CKM.1/RND requires to initialize the (deterministic) random number generator with a secure state, such that it can provide cryptographically good random numbers, that are used for nonces and keying material in the trusted channel protocol,

- the SFR FCS_CKM.4/TCP requires to destroy the session key for the data transport after usage, such that an attacker will not be able to access it to decrypt the transmitted data,

- the SFR FCS_CKM.4/OUT requires to destroy outdated keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, such that they cannot be used for unauthorized access,

- the SFR FCS_CKM.4/RND requires to overwrite the state of the (deterministic) random number generator during initial start-up, such that old sequences of random numbers cannot be reproduced,

- the SFR FCS_CKM.4/KDer requires to destroy the keying material, such that an attacker cannot reconstruct the session key or the MAC3DES key used for data transmission,
- the SFR FCS_CKM.4/MAC3DES requires to destroy the MAC3DES key after use, such that an attacker cannot use it for manipulations,

- the SFR FCS_COP.1/TCPTripleDES requires to use 3DES for encrypting the data before transmission, such that no transmitted data is disclosed,

- the SFR FCS_COP.1/RNDTripleDES requires to use 3DES in the (deterministic) random number generator, such that it can provide cryptographically good random numbers, that are used for nonces and keying material in the trusted channel protocol,

- the SFR FCS_COP.1/KDF requires to use Triple-DES in the key derivation function to derive a 3DES key from A.SymkeyIU necessary for encryption of the data before transmission, and a MAC3DES key for verification of integrity,

- the SFR FCS_COP.1/MAC3DES requires to use MAC3DES to verify the integrity of the data transported through the trusted channel,
- the SFR FCS_RND.1 requires to generate cryptographically good random numbers, that are used for nonces and keying material in the trusted channel protocol,
- the SFR FDP_ETC.2 requires to restrict the access right to data from the A.ProtectedData decrypting and checking engine to the interface unit,
- the SFR FDP_UCT.1/TOE requires to protect A.ProtectedData from unauthorised disclosure,
- the SFR FDP_UIT.1/Data requires to transmit A.ProtectedData in a manner protected from modification, deletion and insertion errors,
- the SFR FTP_ITC.1/TOE requires a trusted channel for transmitting A.ProtectedData to the interface unit,
- the SFR FMT_MSA.2 requires the session key, that is used, to be cryptographically good.

The security objective **O.6** is implemented by the following SFRs:

- the SFR FCS_CKM.1/TCP requires to generate secure nonces used in the trusted channel protocol to identify and authenticate the interface unit,
- the SFR FCS_CKM.1/RND requires to initialize the (deterministic) random number generator with a secure state, such that it can provide cryptographically good random numbers, that are used for the nonces in the trusted channel protocol,
- the SFR FCS_CKM.4/OUT requires (a.o.) to destroy an outdated key A.SymkeyIU, such that it cannot be used for unauthorized access,
- the SFR FCS_CKM.4/RND requires to overwrite the state of the (deterministic) random number generator during initial start-up, such that old sequences of random numbers cannot be reproduced,

- the SFR FCS_COP.1/RNDTripleDES requires to use 3DES in the (deterministic) random number generator, such that it can provide cryptographically good random numbers, that are used for nonces in the trusted channel protocol,

- the SFR FCS_RND.1 requires to generate cryptographically good random numbers, that are used for nonces in the trusted channel protocol,

- the SFR FDP_ACC.2 requires to restrict the access to the A.ProtectedData decrypting and checking engine, such that A.ProtectedData is sent only to the interface unit that is entitled to receive it,

- the SFR FDP_ACF.1 requires to restrict the access to the A.ProtectedData decrypting and checking engine, such that A.ProtectedData is sent only to the interface unit that is entitled to receive it,

- the SFRs FDP_RIP.1 requires that residual information about the outdated keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin be made unavailable, such that it cannot be used for identification of authorization,

- the SFR FIA_UAU.1 requires to authenticate the interface unit before A.ProtectedData is sent to the interface unit,

- the SFR FIA_UID.1 requires to identify the interface unit before A.ProtectedData is sent to the interface unit,

- the SFR FTP_ITC.1/TOE requires a trusted channel for transmitting A.ProtectedData to the interface unit, such that it is verified that the receiver is the interface unit entitled to receive A.ProtectedData,

- the SFR FMT_SMR.1 requires to maintain the role „interface unit“ in the TSF, such that the interface unit can be identified and authenticated,

- the SFR FMT_MSA.1 restricts the right to modify A.SymkeyIU and A.SymkeyAdmin to the administrator, such that A.SymkeyIU is secret, correct and suitable to identify and authenticate the interface unit,

- the SFR FMT_MSA.2 requires the nonces, used in the trusted channel protocol for identification and authentication, to be cryptographically good,

The security objective **O.7** is implemented by the following SFRs:

- the SFR FCS_CKM.1/TCP requires to generate a secure session key for the data transmission,

- the SFR FCS_CKM.1/RND requires to initialize the (deterministic) random number generator with a secure state, such that it can provide cryptographically good random numbers, that are used for nonces and keying material in the trusted channel protocol,

- the SFR FCS_CKM.4/TCP requires to destroy the session key for the data transport after usage, such that an attacker will not be able to access it to decrypt the transmitted data,

- the SFR FCS_CKM.4/OUT requires to destroy outdated keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature, such that they cannot be used any more, especially such that the outdated keys A.SymkeyRNG, A.SymkeyAdmin cannot be used for unauthorized access,

- the SFR FCS_CKM.4/RND requires to overwrite the state of the (deterministic) random number generator during initial start-up, such that old sequences of random numbers cannot be reproduced,

- the SFR FCS_CKM.4/KDer requires to destroy the keying material, such that an attacker cannot reconstruct the session key or the MAC3DES key used for data transmission,
- the SFR FCS_CKM.4/MAC3DES requires to destroy the MAC3DES key after use, such that an attacker cannot use it for manipulations,

- the SFR FCS_COP.1/TCPTripleDES requires to use 3DES for encrypting the data before transmission, such that no transmitted data is disclosed,

- the SFR FCS_COP.1/RNDTripleDES requires to use 3DES in the (deterministic) random number generator, such that it can provide

cryptographically good random numbers, that are used for nonces and keying material in the trusted channel protocol,

- the SFR FCS_COP.1/KDF requires to use Triple-DES in the key derivation function to derive a 3DES key from A.SymkeyAdmin necessary for encryption of the data before transmission, and a MAC3DES key for verification of integrity,
- the SFR FCS_COP.1/MAC3DES requires to use MAC3DES to verify the integrity of the data transported through the trusted channel,
- the SFR FCS_RND.1 requires to generate cryptographically good random numbers, that are used for nonces and keying material in the trusted channel protocol,
- the SFR FDP_ITC.1 requires to restrict the access right to write A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature (i.e. to exchange the keys within the TOE) to the administrator,
- the SFR FDP_UCT.1/TOE requires to protect A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, from unauthorised disclosure,
- the SFR FDP_UIT.1/Keys requires to transmit A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature in a manner protected from modification, deletion and insertion errors,
- the SFR FTP_ITC.1/TOE requires a trusted channel for transmitting A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature to the interface unit,
- the SFR FMT_MSA.2 requires the session key, that is used, to be cryptographically good.

The security objective **O.8** is implemented by the following SFRs:

- the SFR FCS_CKM.1/TCP requires to generate secure nonces used in the trusted channel protocol to identify and authenticate the administrator’s device,

- the SFR FCS_CKM.1/RND requires to initialize the (deterministic) random number generator with a secure state, such that it can provide cryptographically good random numbers, that are used for the nonces in the trusted channel protocol,

- the SFR FCS_CKM.4/OUT requires (a.o.) to destroy an outdated key A.SymkeyAdmin, such that it cannot be used for unauthorized access,

- the SFR FCS_CKM.4/RND requires to overwrite the state of the (deterministic) random number generator during initial start-up, such that old sequences of random numbers cannot be reproduced,

- the SFR FCS_COP.1/RNDTripleDES requires to use 3DES in the (deterministic) random number generator, such that it can provide cryptographically good random numbers, that are used for nonces in the trusted channel protocol,

- the SFR FCS_RND.1 requires to generate cryptographically good random numbers, that are used for nonces in the trusted channel protocol,

- the SFR FDP_ACC.2 requires to restrict the write access to A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, such that only the administrator is entitled to write them,

- the SFR FDP_ACF.1 requires to restrict the write access to A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature, such that only the administrator is entitled to write them,

- the SFRs FDP_RIP.1 requires that residual information about the outdated keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and derived keys be made unavailable, such that it cannot be used for identification or authorization,

- the SFR FIA_UAU.1 requires to authenticate the administrator’s device before accepting new values for A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature,

- the SFR FIA_UID.1 requires to identify the administrator's device before accepting new values for A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature,

- the SFR FTP_ITC.1/TOE requires a trusted channel for transmitting A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature from the administrator's device, such that it is verified that the sender is the administrator entitled to update the keys.

- the SFR FMT_SMF.1 requires the TSF to be capable of performing the security management function "Modification of the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature",

- the SFR FMT_SMR.1 requires to maintain the role „administrator" in the TSF, such that the administrator can be identified and authenticated,

- the SFR FMT_MSA.1 restricts the right to modify A.SymkeyIU and A.SymkeyAdmin to the administrator, such that A.SymkeyAdmin is secret, correct and suitable to identify and authenticate the administrator,

- the SFR FMT_MSA.2 requires the nonces, used in the trusted channel protocol for identification and authentication, to be cryptographically good,

- the SFR FMT_MTD.1.1/Keys requires to restrict the ability to write the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature to the administrator.

The security objective **O.9** is implemented by the following SFRs:

- the SFR FMT_SMF.1 requires the TSF to be capable of performing the security management function "Personalization",

- the SFR FMT_SMR.1 requires to maintain the role „TOE manufacturer" in the TSF, such that the TOE manufacturer can be identified and authenticated,

- the SFR FMT_MTD.1/Pers requires to restrict the ability to exchange program code to the TOE manufacturer,
- the SFR FPT_FLS.1 requires the preservation of a secure state in view of the personalization data,

- the SFRs FPT_TST.1 requires appropriate testing of parts of the TOE in view of the personalization data,
- the SFR FPT_SEP.1 requires to separate the domain for execution of the TSF, such that this domain can be specially protected without interference from other parts of the TOE.

8.2.1.3 OP1 and OP2 versus security assurance requirements

The security objective

- **OP1** “The developer ensures that procedures for the development, design, implementation and testing as required in the ST19WP18-D Security Target [8], which states an augmentation and refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], are applied.”

is covered by the TOE security assurance requirements as stated in Section 5.2 for an EAL3 CC evaluation, in particular by the assurance components ACM (configuration management, which is relevant for development, design, testing, implementation and testing), ALC (Life cycle support, which is also relevant for development, design, implementation and testing by a secure development environment), ATE (Test documentation, which supports appropriate testing) and ADV (Design documentation, which supports in particular design and implementation).

The security objective

- **OP2** “The developer respectively the manufacturer ensures that procedures for the manufacturing, delivery and storage as required in the ST19WP18-D Security Target [8], which refers to the Protection Profiles BSI-PP-0002-2001 [6] and PP/9806 [7], are applied.”

is covered is covered by the TOE security assurance requirements as stated in Section 5.2 for an EAL3 CC evaluation, in particular by the assurance components ACM (Configuration management, which supports in particular the manufacturing procedures but also ensures an appropriate management of delivered and stored items), ALC (Life cylce support, which supports in particular the manufacturing by a secure development respectively manufacturing environment) and ADO (Delivery and Operation, which in particular supports appropriate delivery and storage procedures).

8.2.1.4 Security requirements for the TOE environment and security objectives for the TOE environment

There is the one-to-one correspondence between the security objectives for the TOE environment OE1-OE7 and the security requirements for the non-IT environment SRE1-SRE7. This demonstrates that the combination of all security requirements for the non-IT environment SRE1-SRE7 covers all the security objectives for the environment OE1-OE7. The security objectives for the TOE environment OE1, OE6 and OE7 are additionally supported by the corresponding security functional requirements for the IT environment of the TOE.

The following SFRs for the TOE IT environment correspond to the security objective for the TOE OE1:

- FCS_COP.1/CW_SHA-1,
- FCS_COP.1/CW_DataTripleDES,
- FCS_COP.1/CW_RSASignature,
- FCS_COP.1/CW_MAC3DES.

The intent of the SFRs for the CW listed above is to require that the decrypted A.ProtectedData written to a holographic memory card should be appropriately protected (either encrypted with 3DES and MAC-protected or encrypted with 3DES and RSA signed with the usage of SHA-1). The cryptographical operations should be performed in a way that allows the decryption of the data and the verification of its integrity using cryptographical mechanisms available in the TOE.

The following SFRs for the TOE IT environment correspond to the security objective for the TOE OE6:

- FCS_CKM.1/IU_TCP,
- FCS_CKM.4/IU_TCP,
- FCS_COP.1/IU_TCPTripleDES,
- FCS_COP.1/IU_MAC3DES,
- FDP_UCT.1/IU,
- FDP_UIT.1/IU_Data,
- FDP_ITC.2,
- FTP_ITC.1/IU.

The intent of the SFRs for the IU listed above is to require that the IU is able to set up a trusted channel (FTP_ITC.1/IU) with the TOE using cryptographical primitives

compatible with those available in the TOE (FCS_COP.1/IU_TCPTripleDES, FCS_COP.1/IU_MAC3DES). The generation of session keys should be compatible with the that in the TOE (FCS_CKM.1/IU_TCP). The temporary keys should be also erased in a secure manner (FCS_CKM.4/IU_TCP). Furthermore, the IU should respect the security policy for objects and subjects defined for the TOE (FDP_UCT.1/IU, FDP_UIT.1/IU_Data). When importing the decrypted A.ProtectedData, the IU should take the validity flag which has been set by the TOE into account and react (FDP_ITC.2) in accordance with the end user security policy (e.g., the invalid data can be discarded, the administrator receives a notification, etc.).

The following SFRs for the TOE IT environment correspond to the security objective for the TOE OE7:

- FCS_CKM.1/AU_TCP,
- FCS_CKM.4/AU_TCP,
- FCS_COP.1/AU_TCPTripleDES,
- FCS_COP.1/AU_MAC3DES,
- FDP_UCT.1/AU,
- FDP_UIT.1/AU_Keys,
- FTP_ITC.1/AU.

The intent of the SFRs for the AU listed above is to require that the AU is able to set up a trusted channel (FTP_ITC.1/AU) with the TOE using cryptographical primitives compatible with those available in the TOE (FCS_COP.1/AU_TCPTripleDES, FCS_COP.1/ AU_MAC3DES). The generation of session keys should be compatible with the that in the TOE (FCS_CKM.1/ AU_TCP). The temporary keys should be also erased in a secure manner (FCS_CKM.4/ AU_TCP). Furthermore, the AU should respect the security policy on objects and subjects that is defined for the TOE (FDP_UCT.1/AU, FDP_UIT.1/ AU_Keys).

Thus, the security functional requirements for the TOE IT environment together with the security requirements for the TOE non-IT environment meet the security objectives for the TOE environment.

8.2.2 Dependencies of functional security requirements

The following table gives a short summary of all SFRs and lists their dependencies.

ID	Dependencies	Component	Notes
FCS_CKM.1/TCP	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4 FMT_MSA.2	Cryptographic key generation	Performed by the trusted channel protocol to generate a session key for the data transfer through the trusted channel.
FCS_CKM.1/RND	[FCS_CKM.2 or FCS_COP.1] FCS_CKM.4 FMT_MSA.2	Cryptographic key generation	Initializes the (deterministic) random number generator.
FCS_CKM.4/TCP	[FDP_ITC.1 or FCS_CKM.1] FMT_MSA.2	Cryptographic key destruction	Destroys the session key after the trusted channel is closed.
FCS_CKM.4/OUT	[FDP_ITC.1 or FCS_CKM.1] FMT_MSA.2	Cryptographic key destruction	Destroys the old keys in the TOE after the new keys have been imported.
FCS_CKM.4/RND	[FDP_ITC.1 or FCS_CKM.1] FMT_MSA.2	Cryptographic key destruction	Destroys the state of the deterministic random number generator at start-up.
FCS_CKM.4/KDer	[FDP_ITC.1 or FCS_CKM.1] FMT_MSA.2	Cryptographic key destruction	Destroys the keying material after key derivation.
FCS_CKM.4/MAC3DES	[FDP_ITC.1 or FCS_CKM.1] FMT_MSA.2	Cryptographic key destruction	Destroys the key for the MAC computation after finishing the computation.
FCS_COP.1/SHA1	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Cryptographic operation	Computes the secure hash of enciphered A.ProtectedData for RSA digital signature verification.
FCS_COP.1/DataTripleDES	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Cryptographic operation	Used for decrypting encrypted A.ProtectedData.
FCS_COP.1/TCPTripleDES	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Cryptographic operation	Used for encrypting and decrypting data transported through the trusted channel (see SF.1).
FCS_COP.1/RNDTripleDES	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4	Cryptographic operation	Used as block cipher within the random number generator.

	FMT_MSA.2		
FCS_COP.1/KDF	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Cryptographic operation	Uses Triple-DES within the key derivation functions.
FCS_COP.1/RSASignature	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Cryptographic operation	Used to verify the RSA signature of the SHA-1 hash of enciphered A.ProtectedData.
FCS_COP.1/MAC3DES	[FDP_ITC.1 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Cryptographic operation	Computes the MAC of decrypted A.ProtectedData.
FCS_RND.1	No dependencies	Cryptographic operation	Used for generating random nonces and keying material for the trusted channel protocol.
FDP_ACC.2	FDP_ACF.1	Complete access control	Access to 1. A.ProtectedData decrypting and checking engine, 2. A.SymkeyData, 3. A.SymkeyIU, 4. A.SymkeyRNG, 5. A.SymkeyAdmin, 6. A.PubkeySignature. 7. A.ProtectedData 8. Card identifier
FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	Security attribute based access control	Access to 1. A.ProtectedData decrypting and checking engine, 2. A.SymkeyData, 3. A.SymkeyIU, 4. A.SymkeyRNG, 5. A.SymkeyAdmin,

			6. A.PubkeySignature.
FDP_ETC.2	[FDP_ACC.1 or FDP_IFC.1]	Export of user data with security attributes	Export of decrypted A.ProtectedData
FDP_ITC.1	[FDP_ACC.1 or FDP_IFC.1] FMT_MSA.3	Import of user data without security attributes	Import of 1. A.SymkeyData, 2. A.SymkeyIU, 3. A.SymkeyRNG, 4. A.SymkeyAdmin, 5. A.PubkeySignature.
FDP_RIP.1	No dependencies	Subset residual information protection	Protects old A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin from disclosure.
FDP_SDI.2	No dependencies	Stored data integrity monitoring and action	Protects integrity of decrypted A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature.
FDP_UCT.1/TOE	[FTP_ITC.1 or FTP_TRP.1] [FDP_ACC.1 or FDP_IFC.1]	Basic data exchange confidentiality	Protects decrypted A.ProtectedData, A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin from disclosure.
FDP_UIT.1/Keys	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1]	Data exchange integrity	Protects integrity of A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin, A.PubkeySignature.
FDP_UIT.1/Data	[FDP_ACC.1 or FDP_IFC.1] [FTP_ITC.1 or FTP_TRP.1]	Data exchange integrity	Protects integrity of A.ProtectedData.
FIA_UAU.1	FIA_UID.1	Timing of authentication	Authentication of interface unit and administrator’s device.
FIA_UID.1	No dependencies	Timing of identification	Identification of interface unit and administrator’s device.
FTP_ITC.1/TOE	No dependencies	Inter-TSF trusted	Trusted channel for TOE access

		channel	
FMT_SMF.1	No dependencies	Specification of Management Functions	1. Initialization 2. Personalization 3. Modification of the keys A.SymkeyData, A.SymkeyIU, A.SymkeyRNG, A.SymkeyAdmin and A.PubkeySignature.
FMT_SMR.1	FIA_UID.1	Security roles	Maintains the roles interface unit, administrator, TOE manufacturer.
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	Management of security attributes	Restricts the ability to modify A.SymkeyIU, and A.SymkeyAdmin to the administrator.
FMT_MSA.2	ADV_SPM.1 [FDP_ACC.1 or FDP_IFC.1] FMT_MSA.1 FMT_SMR.1	Secure security attributes	Ensures that only secure values are accepted for security attributes.
FMT_MTD.1/Pers	FMT_SMF.1 FMT_SMR.1	Management of TSF data	Restricts ability to personalize to the TOE manufacturer.
FMT_MTD.1/Keys	FMT_SMF.1 FMT_SMR.1	Management of TSF data	Restricts ability to update keys to the administrator.
FPT_EMSEC.1	No dependencies	TOE Emanation	Limits the physical access to 1. A.ProtectedData, 2. A.SymkeyData, 3. A.SymkeyIU, 4. A.SymkeyRNG, 5. A.SymkeyAdmin, 6. A.PubkeySignature, 7. session key and random numbers (including the state of the deterministic random number generator) used in the trusted channel.
FPT_FLS.1	ADV_SPM.1	Failure with preservation of	Preserves a secure state.

		secure state	
FPT_PHP.3	No dependencies	Resistance to physical attack	Protects the objects under access control from physical manipulation and physical probing.
FPT_TST.1	FPT_AMT.1	TSF testing	Self tests of the security controller ST19WP18-D in the TOE.
FPT_SEP.1	No dependencies	TSF domain separation	Security domain for execution of the TSF.

Table 4: Security functional requirements

The dependencies of FCS_CKM.1/TCP are fulfilled by FCS_COP.1/TCPTripleDES, FCS_CKM.4/TCP and FMT_MSA.2 which requires secure session keys for the trusted channel protocol.

The dependencies of FCS_CKM.1/RND are fulfilled by FCS_COP.1/RNDTripleDES and FCS_CKM.4/RND. The requirement FMT_MSA.2 is not needed, since the administrator generated the secure key A.SymkeyRND.

The dependencies of FCS_CKM.4/TCP are fulfilled by FCS_CKM.1/TCP, and FMT_MSA.2 which requires secure session keys for the trusted channel protocol.

The dependencies of FCS_CKM.4/OUT are fulfilled by FDP_ITC.1 (which addresses the key import by the administrator). The requirement FMT_MSA.2 is not needed, since the administrator generates only secure keys.

The dependencies of FCS_CKM.4/RND are fulfilled by FDP_ITC.1 (which addresses a.o. the key import of A.SymkeyRND by the administrator) and FCS_CKM.1/RND. The requirement FMT_MSA.2 is not needed, since the administrator generates only secure keys.

The dependencies of FCS_CKM.4/KDer are fulfilled by FDP_ITC.1 (which addresses a.o. the import of key material by the administrator from which 3DES and MAC-protection keys are derived) and FCS_CKM.1/TCP. The requirement FMT_MSA.2 requires secure session keys for the trusted channel protocol. FMT_MSA.2 for the key import of A.SymkeyData by the administrator is not needed, since the administrator generates only secure keys.

The dependencies of FCS_CKM.4/MAC3DES are fulfilled by FDP_ITC.1 (which addresses a.o. the key import of A.SymkeyData by the administrator). The requirement FMT_MSA.2 is not needed, since the administrator generates only secure keys.

The dependencies of FCS_COP.1/SHA-1 are fulfilled since the requirements are not needed. The algorithm SHA-1 does not require any key.

The dependencies of FCS_COP.1/DataTripleDES are fulfilled by FDP_ITC.1 (which addresses a.o. the key import of A.SymkeyData by the administrator) and FCS_CKM.4/OUT and FCS_CKM.4/KDer (which destroy key material). The requirement FMT_MSA.2 is not needed, since the administrator generates only secure keys.

The dependencies of FCS_COP.1/TCPTripleDES are fulfilled by FCS_CKM.1/TCP, FCS_CKM.4/TCP and FMT_MSA.2.

The dependencies of FCS_COP.1/RNDTripleDES are fulfilled by FDP_ITC.1 (which addresses a.o. the key import of A.SymkeyRND by the administrator), FCS_CKM.1/RND and FCS_CKM.4/RND. The requirement FMT_MSA.2 is not needed, since the administrator generates only secure keys and the hardware unpredictable numbers are good enough for this kind of seeding of the (deterministic) random number generator.

The dependencies of FCS_COP.1/KDF are resolved by FDP_ITC.1 (which addresses a.o. the import of cryptographic keys by the administrator) and by FCS_CKM.4/KDer (which destroys keying material). FMT_MSA.2 is not needed, since the administrator generates only secure keys and the quality of keying material at the stage of trusted channel initialization is defined by the administrator's random numbers which are cryptographically strong.

The dependencies of FCS_COP.1/RSASignature are fulfilled by FDP_ITC.1 (which addresses a.o. the key import of A.PubkeySignature by the administrator). The requirement FCS_CKM.4 is not needed, since A.PubkeySignature is a public key and therefore its disclosure is acceptable. The requirement FMT_MSA.2 is not needed, since the administrator generates only secure keys.

The dependencies of FCS_COP.1/MAC3DES are fulfilled by FDP_ITC.1 (which addresses a.o. the key import of A.SymkeyData by the administrator from which a 3DES key and a MAC3DES key are derived) and FCS_CKM.4/MAC3DES. The requirement FMT_MSA.2 is not needed, since the administrator generates only secure keys.

The dependencies of FDP_ACC.2 are fulfilled by FDP_ACF.1.

The dependencies of FDP_ACF.1 are fulfilled by FDP_ACC.2 which is hierarchical to FDP_ACC.1. The security functional requirement FMT_MSA.3 is not needed, since the keys stored in the TOE before delivery are cryptographically strong, i.e. a correct

personalization of the TOE is performed, see P2 in Section 3.4. In the end user phase the administrator updates the keys only with cryptographically strong keys, see AE2 in Section 3.2.

The dependencies of FDP_ETC.2 are fulfilled by FDP_ACC.2 which is hierarchical to FDP_ACC.1.

The dependencies of FDP_ITC.1 are fulfilled by FDP_ACC.2 which is hierarchical to FDP_ACC.1. The security functional requirement FMT_MSA.3 is not needed, since the keys stored in the TOE before delivery are cryptographically strong, i.e. a correct personalization of the TOE is performed, see P2 in Section 3.4. In the end user phase the administrator updates the keys only with cryptographically strong keys, see AE2 in Section 3.2.

The dependencies of FDP_UCT.1/TOE are fulfilled by FTP_ITC.1/TOE and FDP_ACC.2 which is hierarchical to FDP_ACC.1.

The dependencies of FDP_UIT.1/Keys and FDP_UIT.1/Data are fulfilled by FDP_ACC.2 (which is hierarchical to FDP_ACC.1) and FTP_ITC.1/TOE.

The dependencies of FIA_UAU.1 are fulfilled by FIA_UID.1.

The dependencies of FMT_SMR.1 are fulfilled by FIA_UID.1.

The dependencies of FMT_MSA.1 are fulfilled by FDP_ACC.2 (which is hierarchical to FDP_ACC.1), FMT_SMF.1 and FMT_SMR.1.

The dependencies of FMT_MSA.2 are fulfilled by FDP_ACC.2 (which is hierarchical to FDP_ACC.1), FMT_MSA.1 and FMT_SMR.1; the assurance requirement ADV_SPM.1 for the TOE is not needed for an EAL3 evaluation. For the part of FMT_MSA.2 that addresses the informal security policy model of the security controller ST19WP18-D in the TOE, the document ADV_SPM.1/ST19WP18-D has already been evaluated during the EAL5+ certification process of the security controller ST19WP18-D.

The dependencies of FMT_MTD.1/Pers and FMT_MTD.1/Keys are fulfilled by FMT_SMF.1 and FMT_SMR.1.

The dependencies of FPT_FLS.1 are fulfilled, since the assurance requirement ADV_SPM.1 is not needed for an EAL3 evaluation. For the part of FPT_FLS.1 that addresses the informal security policy model of the security controller ST19WP18-D in the TOE, the document ADV_SPM.1/ST19WP18-D has already been evaluated during the EAL5+ certification process of the security controller ST19WP18-D.

The dependencies of FPT_TST.1 are fulfilled, since the TOE comprises the software and the hardware, there is no underlying abstract machine the TSF relies upon. Hence the dependency of FPT_TST.1 (TSF self test) upon FPT_AMT.1 (Abstract machine testing) is not relevant here.

There are no dependencies to fulfill for FCS_RND.1, FDP_RIP.1, FDP_SDI.2, FIA_UID.1, FTP_ITC.1/TOE, FMT_SMF.1, FPT_EMSEC.1, FPT_PHP.3 and FPT_SEP.1.

8.2.3 Assurance requirements rationale

The assurance class EAL3 is an established set of mutually supportive and internally consistent assurance requirements, which stands for methodically tested and checked. In particular, security measures are already considered during the design phase and the TOE and the development process are thoroughly investigated. This assurance level is considered to be suitable and sufficient to meet the security objectives and corresponds well to the assumed low level of experience, opportunity and resources of an attacker.

The complete EAL3 assurance package was chosen, so all dependencies are fulfilled.

8.2.4 Mutual support and internal consistency

The following part of the security requirements rationale shows that the set of security requirements for the TOE consisting of the security assurance requirements (SARs) and the security functional requirements (SFRs) together forms a mutually supportive and internally consistent whole.

The analysis of the TOE's security requirements with regard to their mutual support and internal consistency demonstrates:

The assurance class EAL3 is an established set of mutually supportive and internally consistent assurance requirements.

The TOE shall meet the set of the security assurance requirements encompassing the evaluation assurance level EAL3 (methodically tested and checked) according to Common Criteria 2.2 Part 3 [4]. The set of these assurance requirements are mutually supportive and internally consistent as all dependencies are satisfied and no inconsistency appears.

The dependency analysis in Section 8.2.2 for the security functional requirements shows that the basis for mutual support and internal consistency between all defined functional requirements is satisfied. All dependencies between the chosen functional components are analysed, and non-dissolved dependencies are appropriately explained.

The following additional reasons support consistency and mutual supportiveness of the SFRs:

The chosen SFRs of class FCS implement the cryptographic algorithms as required by the TOE specification.

The chosen SFRs of classes FIA, FDP and FTP support the access control policy **SFP_access_control** as defined in Section 5.1.

The chosen SFRs of class FMT support the secure management of TSF data in a way, which is consistent to the policy **SFP_access_control**.

The SFRs of all these classes (FCS, FIA, FDP, FTP, FMT) together provide the TOE services as defined in the TOE description.

The remaining SFRs, chosen from class FPT and FPR define low level protection of the TOE against any attempt to bypass the security policy **SFP_access_control** or the services defined in the specification.

Any inconsistencies between security functional and security assurance requirements do not arise and the chosen assurance components are adequate for the functionality of the TOE (see 8.2.3). So the assurance requirements and security functional requirements support each other and there are no inconsistencies between the goals of these two groups of security requirements.

8.2.5 Strength of function level rationale

The minimum strength level for the TOE security functional requirements is claimed as SOF-medium. This is suitable to meet the security objectives for the TOE since these objectives are appropriate to avert the threats, where attackers possessing a low level of experience, opportunity and resources are assumed. The chosen level also fits with the selected EAL3 assurance level.

8.3 TOE summary specification rationale

This section demonstrates that the set and combination of the TOE security functions is suitable to satisfy the identified security functional requirements. In particular, it shows that each security function is related to at least one functional requirement and vice versa. The detailed description and analysis of the TOE security functions in

Section 6.1 already showed that these functions work together well and support each other and that no inconsistencies exit.

The section also includes an appropriate strength of functions rationale and lists the assurance measures of the developer to meet the security assurance requirements. As already stated the assurance level EAL 3 is considered to be suitable for the TOE and the deliverables listed in Section 6.2 are considered to be suitable and sufficient to fulfill the assurance requirements.

8.3.1 Security requirements - security functions

The following table shows, which security functions of the TOE support which security requirements for the TOE. The table shows, that every security requirement is supported by at least one security function and that every security function supports at least one security requirement.

For confidentiality reasons, the explanations are not reproduced here.

Security Requirements vs. Security Functions	SF.1	SF.2	SF.3	SF.4	SF.5	SF.6	SF.7	SF.8	SF.9	SF.10
FCS_CKM.1/TCP	X							X	X	
FCS_CKM.1/RND								X		
FCS_CKM.4/TCP	X									
FCS_CKM.4/OUT		X				X				
FCS_CKM.4/RND								X		
FCS_CKM.4/KDer									X	
FCS_CKM.4/MAC3DES	X		X							
FCS_COP.1/SHA-1			X						X	
FCS_COP.1/DataTripleDES			X							
FCS_COP.1/TCPTripleDES	X									
FCS_COP.1/RNDTripleDES								X		
FCS_COP.1/KDF	X		X						X	
FCS_COP.1/RSASignature			X							
FCS_COP.1/MAC3DES	X		X							

FCS_RND.1								X		
FDP_ACC.2	X				X	X	X			X
FDP_ACF.1	X				X	X	X			X
FDP_ETC.2	X				X	X	X			X
FDP_ITC.1	X				X	X	X			X
FDP_RIP.1	X	X	X							X
FDP_SDI.2	X				X	X	X			X
FDP_UCT.1/TOE	X				X	X	X			X
FDP_UIT.1/Keys	X				X	X	X			X
FDP_UIT.1/Data	X				X	X	X			X
FIA_UAU.1	X				X	X	X			X
FIA_UID.1	X				X	X	X			X
FTP_ITC.1/TOE	X									
FMT_SMF.1		X		X		X				
FMT_SMR.1				X		X	X			
FMT_MSA.1	X				X	X	X			X
FMT_MSA.2	X							X	X	
FMT_MTD.1/Pers	X			X	X	X	X			X
FMT_MTD.1/Keys	X				X	X	X			X
FPT_EMSEC.1										X
FPT_FLS.1		X								X
FPT_PHP.3										X
FPT_TST.1		X								
FPT_SEP.1										X

Table 5: Security Requirements vs. Security Functions

8.3.2 Strength of function rationale

The level of the strength of the TOE's security functions is claimed as SOF-medium. As already presented in Section 6.1 the following TOE security functions based on probabilistic or permutational mechanisms are identified:

- SF8: The generation of random numbers (E.4 from AIS20 [10], recursive call of 3DES with 2 keys) using SHA1 hashed ST19WP18-D hardware unpredictable numbers and an additional secret key A.SymkeyRND.
- SF1: Data and subject authentication during the initialization and usage of the trusted channel. For the authentication the following mechanisms are used:
    - Message Authentication Code (MAC) is the 3DES CBC encryption algorithm with two keys and a constant IV. The last 64-bit ciphertext block is used as the resulting MAC value. See also [8].
    - 128-bit challenges used to guarantee unpredictability of input and freshness of established keys according to [9]. The generator of random numbers (see SF8) is used to produce the challenges on the side of the TOE.

  Note that these mechanisms of the security function SF1 are also used by security functions SF6 and SF7.
- SF3: Verification of the integrity and authenticity of encrypted A.ProtectedData. The following mechanisms are used:
    - Message Authentication Code (MAC) is the 3DES CBC encryption algorithm with two keys and a constant IV. The last 64-bit ciphertext block is used as the resulting MAC value. See also [8].
    - The SHA-1 hash function with 160-bit outputs (described in the ST19WP18-D Security Target [8]) is used in the RSA-PSS signature method to bind the encrypted A.ProtectedData to the signature.

8.3.3 Assurance requirements - Assurance measures

Assurance measure		Assurance components	Notes
M.1	Configuration management	ACM_CAP.3	Authorization control
		ACM_SCP.1	TOE configuration management coverage
M.2	Delivery and operation	ADO_DEL.1	Delivery
		ADO_IGS.1	Installation, generation and start-up procedures
M.3	Informal functional specification	ADV_FSP.1	Informal functional specification
M.4	Security specific high level design	ADV_HLD.2	Security specific high level design
M.5	Informal evidence of representation correspondence	ADV_RCR.1	Informal evidence of representation correspondence
M.6	Handbooks	AGD_ADM.1	Administrator guidance
		AGD_USR.1	User guidance
M.7	Life cycle support	ALC_DVS.1	Development security
M.8	Test documentation	ATE_COV.2	Coverage analysis
		ATE_DPT.1	High level test design
		ATE_FUN.1	Functional tests
		ATE_IND.2	Independent tests
M.9	Vulnerability assessment	AVA_MSU.1	Examination of guidance
		AVA_SOF.1	Strength of TOE security functions
		AVA_VLA.1	Developer vulnerability analysis

Table 6: Assurance requirements vs. assurance measures

8.4 PP claims rationale

No conformation to any protection profile is provided.

9 References

1. „IT-Sicherheit geht alle an!“ Tagungsband zum 9. Deutschen IT-Sicherheitskongress, Bundesamt für Sicherheit in der Informationstechnik, 2005
2. Common Criteria for Information Technology Security Evaluation. Part 1: Introduction and general model. Version 2.2. January 2004
3. Common Criteria for Information Technology Security Evaluation. Part 2: Security functional requirements. Version 2.2. January 2004
4. Common Criteria for Information Technology Security Evaluation. Part 3: Security Assurance Requirements. Version 2.2. January 2004
5. Common Methodology for Information Technology Security Evaluation, Evaluation Methodology. Version 2.2. January 2004
6. Smartcard IC Platform Protection Profile. Version 1.0. BSI-PP-0002-2001. July 2001
7. Protection Profile Smartcard Integrated Circuit. Version 2.0. PP/9806. September 1998
8. ST19WP18-D Security Target SMD_ST19WP18D_ST_06_001_V01.00, February 2006
9. ISO/IEC 11770-2:1996(E), International Standard. Information technology – Security techniques – Key management – Part 2: Mechanisms using symmetric techniques, 1996
10. Application Notes and Interpretation of the Scheme (AIS), AIS 20, Version 1, Functionality classes and evaluation methodology for deterministic random number generators, BSI, December 1999
11. Common Criteria Protection Profile electronic Health Card (eHC), BSI, Version 1.02, December 12, 2005
12. PKCS #1 v2.1: RSA Cryptography Standard, RSA Security Inc., June 2002