SafeGuard Enterprise - Device Encryption

Document Version:	1.00.00
Document Type:	Security Target (public version)
Project Id:	SGN 5.60
File Name:	SGN_EAL4_ST_L.docx
Author(s):	R. Reinl, J. Schneider, C. Tobias, A. Wenzel
Office / Company:	Utimaco Safeware AG

Abstract: This document contains the Security Target for the Common Criteria certification of SafeGuard Enterprise - Device Encryption, version 5.60
Certification ID: BSI-DSZ-CC-0553

Table of Contents

1 Document Information

1.1 Owner / Master Location

Owner of this document is Joachim Schneider (JOS). The location of the master copy is the server of the Subversion configuration management tool located in Munich at

https://svn-munich/svn/SGNCertification/trunk/sw/doc/SGN_EAL4/SGN_EAL4_ST_L.docx.

1.2 Change History

Version	*Author*	*Date (finished)*	*Description*
1.00.00	JOS	02.05.2012	First Version

1.3 Distribution & Approval History

Version	*Distributed to / approved by*	*Date distributed*	*Date approved*
1.00.00	SRC GmbH / BSI	02.05.2012	02.05.2012

2 ST Introduction

The chapter *ST Introduction* is divided into the following sections:

ST Identification – contains an identification of the TOE,

ST Overview – contains a short overview over the TOE's functions,

CC Conformance – contains the claims for the conformance of the Security Target to the CC.

2.1 ST Identification

This document provides the Security Target as basis for the evaluation of the software product SafeGuard Enterprise – Device Encryption for Windows.

The Target of Evaluation (TOE) is identified as:

SafeGuard Enterprise – Device Encryption, Version 5.60 for Microsoft Windows XP Professional and Microsoft Windows 7

2.2 ST Overview

2.2.1 SafeGuard Enterprise Security Suite

SafeGuard Enterprise is a modular security suite that meets all current and future demands on data security. No matter where information is saved, or who it is being exchanged with, SafeGuard Enterprise secures data on mobile and fixed computing devices, on removable media, servers and in e-mails.

Figure 1: Architecture of SafeGuard Enterprise Security Suite

SafeGuard Enterprise consists of the following individual modules:

- SafeGuard Management Center
- SafeGuard Device Encryption (the TOE mentioned here)
- SafeGuard File and Folder Encryption (available in version 6.0)
- SafeGuard Configuration Protection
- SafeGuard Data Exchange

2.2.2 SafeGuard Management Center

SafeGuard Management Center provides the customer with complete security and management control over all the connected devices and users. This is the central management console of SafeGuard Enterprise. It works in conjunction with the SafeGuard Device Encryption module to deliver the highest levels of data security and performance.

With SafeGuard Management Center a security policy can be enforced throughout the organization. Definitions, implementation, and control are centralized:

- A role-based management system enables the definition of security rules for a large number of user groups, organizational units, and devices. The user management works in interconnection with the user management of Windows Server Active Directory.
- Security rules can be distributed to endpoint devices quickly and conveniently.
- Reporting and auditing features provide up-to-date information about the security status of all devices.

2.2.3 SafeGuard Device Encryption

SafeGuard Enterprise Device Encryption prevents unauthorized users from access to the clear text of data stored on mobile and stationary endpoint devices. This is achieved by encryption of the data. Encryption is completely transparent to users. If the end device falls into the wrong hands, the stored plaintext data is unreadable even if the hard disk is removed. SafeGuard Device Encryption provides protection for built-in storage media like hard disks as well as for mobile data media, such as USB memory sticks, memory cards (e.g. SD/MMC) and Compact Flash. This ensures that data stored on mobile media is secured by encryption during transport.

This module of SafeGuard Enterprise is the Target of Evaluation (TOE) discussed in this Security Target document.

By its encryption capabilities, this module is one of the security enforcing parts of the SafeGuard Enterprise security suite, therefore it is exposed to security evaluation.

Each module of SafeGuard Enterprise supports the overall security policies of the security suite. On the other hand, each module is a separately developed product; all of them combined by a common security policy database and commonly used interfaces.

2.3 CC Conformance

The *Security Target* is structured according to the general rules listed in Part 1 of the Common Criteria [CC1].

The *TOE Security Assurance Requirements* claim to be conformant to Part 3 of the Common Criteria [CC3],

The TOE *Functional Requirements* for the TOE claim to be conformant to those in Part 2 of the Common Criteria [CC2] extended with Security Functional Requirement FCS_RND.1 (taken from [AIS31]).

The evaluation is based upon

[CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 2.3, August 2005

[CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 2.3, August 2005

[CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 2.3, August 2005

The *Functional Requirements* claimed for the TOE are described in section 6.1 of this document.

The *Evaluation Assurance Level* claimed by the evaluation is

EAL4 (Evaluation Assurance Level 4)

All individual assurance components for measuring the achieved assurance are compliant to Part 3 of the Common Criteria [CC3]. The assurance components comprise all required components for EAL4 according to [CC3]. They are all listed in section 6.1.1 of this document.

The TOE does not claim to be conformant to any *PP*.

The assurance of the *Minimum Strength of Function* (SOF) is claimed to be

SOF-medium

according to Part 3 of the Common Criteria [CC3].

3 TOE Description

The chapter *TOE Description* is divided into the following sections:

General Description – contains a basic description of the TOE,

TOE Components – contains a listing of the main components of the TOE,

TOE Hardware and Software Environment – contains a description of the technical IT environment, where the TOE is intended to be installed.

TOE Boundaries – describes briefly the external interfaces of the TOE,

3.1 General Description

SafeGuard Enterprise – Device Encryption is a part of the SafeGuard Enterprise security suite.

One major purpose of SafeGuard Enterprise Device Encryption is the sector based device encryption for preventing unauthorised individuals from accessing the clear text of the data stored on magnetic and solid state mass storage devices for PCs, named “block devices” in the following. This includes PCs’ hard disks and their partitions as well as portable data storage devices (USB sticks, external hard disks, floppy disks etc.). It does not include devices handled by specific file system drivers, like network devices, CD-ROMs, CD-R/W, DVD and some MO devices.

The administration of SafeGuard Enterprise – Device Encryption is done with the help of other components of SafeGuard Enterprise. A centralized database holds the security policies, user roles, device properties, key rings and other configuration data. This configuration data is forwarded over a network connection to the TOE on each client PC. This centralized database is maintained by the SafeGuard Management Center, which is also part of the SafeGuard Enterprise security suite, but not part of this evaluation.

For this specific evaluation, the main focus is on the device encryption mechanisms, namely the sector-by-sector encryption of data media by the SafeGuard Enterprise – Device Encryption (the TOE). Where sector-by-sector encryption is not possible or not desirable, all data on the media is encrypted on a file-by-file basis. For the evaluation, the existence of an appropriate administration console and server is assumed. This is modelled by the definition of a remote connection between the TOE and an administration server, over which the configuration data is imported into the TOE.

3.2 TOE Components

The Target of Evaluation (TOE) consists of

(i) the installable program code of the Device Encryption client for SafeGuard Enterprise Version 5.60, English program version. The program code is a part of SafeGuard Enterprise, delivered on the SafeGuard Enterprise product DVD, identified as

SafeGuard® Enterprise 5.60.0.192

Application for Windows XP / Vista / Windows 7

The following parts of the installed programs implement the security functionality of the TOE:

(a) the system kernel of SafeGuard Enterprise – Device Encryption (including modified master boot record and POA code),

(b) the drivers needed for encrypting and decrypting user data,

(c) the parts needed for installation of system kernel and initial hard disk encryption,

(d) the communication interface to the SafeGuard Management Center.

(ii) the installation manual of SafeGuard Enterprise – Device Encryption, called ”[SafeGuard® Enterprise Version 5.60] – [Installation Manual]” contained as Acrobat PDF files on the product DVD.

(iii) the Administration Guide for installing, administering and maintaining SafeGuard Enterprise, called “[SafeGuard® Enterprise Version 5.60] – [Administrator Help]” contained as Acrobat PDF files on the product DVD.

(iv) the User’s Guide for operating SafeGuard Enterprise and SafeGuard Enterprise – Device Encryption, called “[SafeGuard® Enterprise Version 5.60] – [User Help]” contained as Acrobat PDF files on the product DVD.

(v) the User's Guide Enhancement for secure operation, called ”[SafeGuard® Enterprise] – [Manual for certification compliant operation]” contained as Acrobat PDF files on the product DVD.

Note1:

The SafeGuard Enterprise – Device Encryption product DVD does also contain a German, French and Japanese program version which is not within the scope of this evaluation and therefore not explicitly tested. Since however the program components constituting SafeGuard Enterprise’s functionality are identical for all language versions (differences are made up only by language files and/or resource DLLs containing text and/or bitmaps), the evaluation results for the security objectives, the functional and assurance requirements may also apply to the German, French and Japanese program version.

Note2:

The TOE consists only of the SafeGuard Enterprise – Device Encryption. The SafeGuard Enterprise Management Center is not part of the TOE, but provides the required

administration and maintenance functions for the TOE. However, due to a well defined management interface, even other systems may act as management components for the TOE.

3.3 TOE Hardware and Software Environment

3.3.1 Hardware Requirements

The TOE runs on personal computer systems with following minimum requirements:

- microprocessor Intel Pentium IV with at least 1.3 GHz speed or compatible device, with 32-bit or 64-bit internal operation, suitable for Windows XP (32-bit only) and/or Windows 7 (32 or 64 bit),
- minimum system RAM of 1 GB (2 GB recommended for Windows 7),
- 5 GB free hard disk space (minimum)
- DVD-ROM drive for installation,
- USB 2.0 port (if a USB Token device is used for authentication).
- USB Token (optional), e.g. Aladdin eToken or RSA SecurID Token
- Smart Card Reader (if smart cards are used for authentication)
- Smart Card (optional), several Java Card, MultOS and ISO 7816 cards are supported

3.3.2 Software Requirements

Operating System

The version of SafeGuard Enterprise – Device Encryption under this evaluation is provided for the following operating systems:

Microsoft Windows XP 32-bit Professional Edition Service Pack 3
Microsoft Windows 7 32-bit Ultimate Edition
Microsoft Windows 7 64-bit Ultimate Edition

For all operating systems, the international versions for support of Western character sets are applied.

SafeGuard Enterprise – Device Encryption works with all available file systems under Windows XP/Windows 7: FAT, FAT32, NTFS4, and NTFS5 (EFS).

Application Software Requirements

The TOE is compatible with all application software released for the mentioned operating system platform. However, application software which does not use the respective Application Programming Interface of the OS platform for disk access, circumventing some layers of the disk access system, may read encrypted sectors from the disk and therefore may not recognise the file structure on the disk correctly. Such software may also write plain text data directly onto a protected device. Then this data is not protected by the TOE against unauthorised disclosure.

In practice, such software is not known to the vendor, except for special hard disk repair and copy functions. Using such software for hard disk repair and copy functions while the TOE is installed is not advised, as this also may - in extreme consequence - damage the TOE installation and the user data.

3.3.3 Connectivity Aspects

Administration Network Connection

The PC, where the TOE as a part of SafeGuard Enterprise shall work, must have a network connection to an administration server. This server accesses the SafeGuard database, which holds the security policies and administration data. The database is in turn maintained by the SafeGuard Management Center.

The connection is done via a web server interface: the server provides a web server socket that the client connects to in order to retrieve its administration and configuration data.

A copy of data relevant for the PC with the TOE installed is stored locally on the PC to enable operation of this PC in case a connection to the administration server is not possible.

Whenever a connection to the administration server can be established, the administration and configuration data is periodically synchronized with the administration server.

The data connection between SafeGuard Enterprise Device Encryption and SafeGuard Enterprise Server has to be secured by a Secure Socket Layer (SSL) resp. Transport Layer Security (TLS) connection provided by the IT environment.

Further Connectivity Aspects

The PC with the TOE is normally connected to a network or might even be directly connected to another system using a cable connection (e.g. serial).

In these cases it must be observed that the TOE only protects local devices, and that there is no encryption of remote drives in networked environments.

Also, the protection mechanism may not be effective when the secured PC is operated while connected to another computer system and parts of the PC's built-in or removable devices are accessible to other users or programs (via shared partitions/drives/volumes, directories or files) using this connection. In this case, any user having access to those shares has access to the plain text data stored on it.

For these reasons the threats defined for the TOE are restricted to unauthorised accesses to plain text data by unauthorised users on PCs not in an operational state, i.e. the unauthorised individual tries to access data by either trying to set the PC into operation, removing the hard disk from the PC, or taking an encrypted removable device and examining the device separately.

Additionally, attention must be paid to the fact that, when the PC - with the TOE installed on it - is operated in connection to any other computer system, it might be possible for unauthorised individuals to manipulate the TOE in a way that its security functionality can be circumvented or deactivated (e.g. by installing “Trojan Horse”-type programs/scripts). Therefore no partition-/drive-/volume-, directory- or file-shares shall be defined on a PC secured by the TOE.

When the TOE is operated in a network with connection to the Internet, a correctly installed and maintained firewall system shall be established to prevent access to the protected PC's hard disk(s) and memory by unauthorised individuals from outside.

3.4 TOE Boundaries

3.4.1 TOE Representation

SafeGuard Enterprise – Device Encryption is a pure software product. It consists of applications, drivers, native machine code and data files provided for running under Microsoft Windows operating systems on industry standard PCs.

3.4.2 TOE External Interfaces

The TOE provides the following external interfaces:

- An interface to the BIOS of the underlying PC for the processing of hard disk read/write accesses during real mode operation.
- An interface to the block device driver of FreeBSD for the processing of hard disk read/write accesses during POA.
- An interface to the storage driver stack of the underlying operating system (Windows XP, Windows 7) for the processing of hard disk read/write accesses during protected mode operation.
- An interface to the administration server (not part of the TOE), where administration data and configuration data are forwarded to the TOE on the secured PC.
- An interface to the operating system for the synchronisation of the passwords between OS and TOE.
- An interface to the user: this is a graphical user interface, where the user can read data on the screen and can enter data via keyboard and other input devices (e.g. mouse). This interface is used for user identification/authentication and for TOE administration support.
- Optionally, an interface to a smart card reader or USB token. This interface is used for user identification/authentication if the TOE is configured to use a token or smart card.

3.5 TOE Delivery

The delivery of the TOE is secured in a way that any user can determine the authenticity of the software package received.

The installation files (.msi) of the TOE are digitally signed with a VeriSign class 3 Code Signing Certificate. This enables customers to verify the origin, integrity and authenticity of the TOE.

This feature is explained in detail in the installation guide delivered together with the product to make the user aware of the existence of such a check.

There are both 32-bit and 64-bit versions of the installation files. Although most components are identical for both these files and all components are built from the same source tree, a few components need to be compiled with different parameters for a 64-bit operating system because the operating system interfaces require a 64-bit executable.

The TOE and supporting infrastructure are installed from the installation files on the product DVD listed in the following table. Only one of the first two files named *SGNClient…* installs the TOE proper (depending on operating system platform). However, verification information for all five is available to ensure the TOE can be operated in a verified environment.

File	Comment	SHA256 Hash
SGNClient.MSI	32-bit client	3d76caf9fff25e561a39aa0104f45fc52735a85733ade3c88a60586d36904d8d
SGNClient_x64.MSI	64-bit client	6c878420c1426cf2659bc8611f07f4b26fd8c4aeebfadf9638b233df5978df35
SGNManagementCenter	Management Console	420b8b72c1da328d91670338871e1a4575891c2d6e63e87dde1720cf5ae2c628
SGNServer.MSI	Server	ffd0441c3e63c8912d1cb6e1e68a6b3a3ca19d3abc0b74761aa2a966812756eb
SGxClientPreInstall.MSI	Runtime Environment Update	e5ed69917e32bb5e570508cb51635686822070d96da47c2a397d601a68960d6a

The hashes can be verified with various hashing tools; they were created using the FSUM tool available at www.slavasoft.com/fsum. This tool can be used for verification as well. For convenience, use the following text block as a file.

```
;paste following lines into a text file and run fsum -c <text file name> to verify
3d76caf9fff25e561a39aa0104f45fc52735a85733ade3c88a60586d36904d8d ?SHA256*SGNClient.msi
6c878420c1426cf2659bc8611f07f4b26fd8c4aeebfadf9638b233df5978df35 ?SHA256*SGNClient_x64.msi
420b8b72c1da328d91670338871e1a4575891c2d6e63e87dde1720cf5ae2c628 ?SHA256*SGNManagementCenter.msi
ffd0441c3e63c8912d1cb6e1e68a6b3a3ca19d3abc0b74761aa2a966812756eb ?SHA256*SGNServer.msi
e5ed69917e32bb5e570508cb51635686822070d96da47c2a397d601a68960d6a ?SHA256*SGxClientPreinstall.msi
```

4 Security Environment

The chapter *Security Environment* is divided into the following sections:

Subjects, Objects and Operations – contains a definition of subjects, objects and operations

Secure Usage Assumptions – contains the assumptions made for the operation of the TOE,

Threats – contains a description of the threats averted by the TOE and the environment,

Organisational Security Policies – contains a description of the distinction of users in the TOE.

4.1 Subjects, Objects and Operations

To simplify the definition of assumptions and of threats countered by the TOE, a definition of subjects and objects is preceded.

4.1.1 Subjects

Subjects relevant for considering the security of the TOE are:

<S.AUTH> Authorised individuals, i.e. persons who are authorised by the security policy to have access to the boot device of a PC with the TOE installed and - optionally - to other devices built in the PC or temporarily attached to a PC (portable devices).
The following security attributes are assigned to that subject: User name, password.

<S.UNAU> Unauthorised individuals, i.e. persons who are not authorised by the security policy to have access to the boot device of a PC with the TOE installed and to other devices built in the PC or temporarily attached to a PC (portable devices).
The following security attributes may also be assigned to that subject: None.

4.1.2 Objects

Objects relevant for considering the security of the TOE are mainly data objects (abbreviated with D.):

<D.USER> Plain text user data contained in any device secured by the TOE; plain text user data encloses data files, program files, operating system files and file system information on a block device supported by the TOE.

<D.TSF> TSF data, i.e. management data required for the administration and operation of the TOE. This data includes information about authorised

users, their access rights and encrypted keys for device encryption. TSF data is stored on a specific part of the boot partition of the secured PC. This is a copy of administration data received from the remote administration server.

4.1.3 Access Operations

The TOE policy is to prevent unauthorised users from access to information by using encryption methods. Information, called "plain text" in the further document, is hidden by being encrypted into “cipher text”. The following access operation is defined to specify the threats and the security policy of the TOE:

<ACC.SUB> Substantial Access,
means, that an individual – authorised (<S.AUTH>) or unauthorised (<S.UNAU>) – is accessing, i.e. reading plain text information on a secured device, which is part of the object “user data” (<D.USER>).

<ACC.ADM> Administrative Access
means, that an unauthorised individual or a process – (<S.UNAU>) – is modifying the security configuration of the TOE, i.e accessing <D.TSF>.

Furthermore it is defined that substantial access by unauthorised individuals (<S.UNAU>) is only averted when the PC, where the device is built in or attached to, is not in operational state, or when the (portable) device is detached from any PC. The PC is in operational state, after an authorised (<S.AUTH>) individual has performed login, until the moment, where the operating system has been shut down and the PC has been physically switched off. It is important to mention that the PC remains in operational state when the user invokes any screen/keyboard locking function or any suspend mode provided by the operating system or by the PC's BIOS, which does on its resume not require a reboot from the master boot record. During operational state, the plain text user data on the hard disk is always accessible, because the TOE's encryption/decryption functionality is active or can be used without any further authentication. A portable device defined as encrypted is not secured, as long as it is attached to a PC secured by the TOE, where an authorised (<S.AUTH>) individual has performed login. The device is secured, if it is not attached to any PC or if it is attached to a PC, which is not secured by the TOE, or where an unauthorised individual (<S.UNAU>) has performed login.

4.2 Secure Usage Assumptions

In this section several assumptions (or requirements to use the TOE) are described. The first sections describe the hardware and software environment, where SafeGuard Enterprise – Device Encryption is designed to fulfil its security functions. The next sections describe the physical, personnel, organisational and connectivity aspects which have to be regarded to operate SafeGuard Enterprise – Device Encryption in a way that the TOE’s security can be guaranteed.

SafeGuard Enterprise – Device Encryption is a software product installed on a PC to prevent unauthorised access to user data stored on storage devices. Only authorised individuals may use the computer (start/boot the operating system from an encrypted device, especially from the hard disk).

The following assumptions are made to guarantee the TOE's security:

<A.INST> Installation Options
It is assumed that the TOE is properly installed and configured regarding the required settings for the security attributes. These settings are listed in detail in the corresponding requirement <R.INST> in section 6.4 of this document.

<A.PASSW> Non-Disclosure of Passwords and PINs
It is assumed that measures are taken to ensure all authorised individuals <S.AUTH> protect their passwords and PINs in a way that they are not disclosed to unauthorised individuals <S.UNAU>. This includes protection against password recording using hardware devices or software tools.

<A.DIRECT> Avoiding Inappropriate Application Software
It is assumed that untrusted software, which does not use the respective Application Programming Interface of the OS platform for disk access but directly accesses the hard disk by circumventing layers of the disk access system, is not placed on the PC's hard disk and not executed while the computer is operated.

<A.ADMIN> Trusted administrator and administration tools
It is assumed that the administrators responsible for the administration server can be trusted and that the administration server is operated in a secure environment.

<A.USER> Adequate User behaviour
It is assumed that authorised users do not actively or negligently compromise the security of the computer on which the TOE is installed. Examples for such compromising actions would be:

- Placing malicious software (like programs containing viruses or Trojan horses) on the computer,
- modifying the TOE program or data files,
- modifying the hard disk with tools circumventing the TOE transparent encryption interface or
- leaving a computer secured by the TOE unattended while being in operational state.

<A.PHY_CTL> The computer secured by the TOE shall not fall under temporary and undetected physical control of an attacker. If a token or a smart card is used for authentication, the assumption extends to the token or the smart card and smart card reader.

<A.TOKEN> If a token or smart card is used for user authentication it is assumed that the device implements secure storage of the user's private key through its hardware and firmware/operating system, and that it requires a PIN before any operation using this key can be performed..

4.3 Threats

The following threats are claimed be averted by the TOE:

<T.ACCESS> An unauthorised individual <S.UNAU> attempts to perform a substantial access <ACC.SUB> to any data stored on encrypted devices <D.USER>. This attack is expected to be performed when the PC is not in operational state.
("Substantial access" means reading clear text of the data; "any data" means data files, program files, operating system files and file system information).

<T.REMADMIN> An unauthorised individual or process <S.UNAU> attempts to change the TOE security configuration (<ACC.ADM>):

- Changing the protection status of the TOE or modifying other TSF data) via the configuration network interface, where the administration server is normally connected. This means the unauthorised individual or process is pretending to be an administration server.
- Changing the password of an authorized individual S.AUTH (local password change).

<T.KEYGEN> An unauthorised individual <S.UNAU> succeeds in guessing cryptographic keys due to weak keys generated by the TOE's key generation mechanisms.

4.4 Organisational Security Policies

The Security Objectives of the TOE (chapter 5) are only derived from the identified threats (section 4.3) together with assumptions (section 4.2). There are no additional organisational security policies defined.

5 Security Objectives

The chapter *Security Objectives* is divided into the following sections:

TOE Security Objectives – contains the security objectives for the TOE,

Security Objectives for Environment – contains the security objectives for the environment.

The *Security Objectives* can be directly traced back to the *Threats* defined in section 4.3. Nevertheless, the description of the *Security Objectives* contains additional information to indicate how the security problem (*Threat*) is addressed by the TOE. This is to provide a clear link to understand the *TOE Functional Requirements* and the *IT Environment* and *Non-IT Environment Requirements*.

5.1 TOE Security Objectives

The TOE is designed to prevent unauthorised users from access to data and programs on PC hard disk partitions.

The *TOE's Security Objectives* are as follows:

<O.ACCESS> Unauthorised individuals <S.UNAU> shall not be able to perform any substantial access to user plain text data stored on devices defined as encrypted by the TOE <D.USER>. This attempt is expected to be performed when the PC is not in operational state.
Solution:
This user data is protected by a TSF which encrypts the user data whenever being written onto the device. Authorised individuals are identified by checking their respective password before the operating system is loaded. The latter function provides the cryptographic key necessary to access (decrypt and encrypt) the data stored on the protected devices.

<O.MANAGE> Unauthorised individuals <S.UNAU> shall not be able to perform particular TOE management operations.

Solution:
The confidentiality of cryptographic key material and passwords and the integrity revocation lists transmitted between the remote administration server and the TOE is secured by cryptographic operations. Authorised users can change their password after entering their current one.

<O.KEYGEN> Unauthorised individuals <S.UNAU> shall not be able to determine cryptographic keys by analysis of the key generation mechanism.

Solution:
For key generation of local machine keys and device encryption keys of the local block devices the TOE uses a random number based key generator generating unpredictable results.
If other keys are used by the TOE (e.g. device encryption keys for mobile devices) it is assumed, that these keys are also generated by an appropriate key generation mechanism.

5.2 Security Objectives for Environment

The *Security Objectives for Environment* are as follows:

<OE.SERVER> An administration server is connected to the installed TOE for transferring the security configuration (TSF data) to the TOE. The TOE and the administration server communicate using a secure SSL connection that is configured as described in the manual for certification compliant operation [CERTCOM]. In particular, the SSL connection is configured such that it provides a strong server authentication and strong encryption and integrity protection of all transmitted data.
The SGN Management Center fulfils this requirement if the TOE is correctly installed and configured according to the installation requirements.
Changing configuration data on the remote administration console shall only be possible for authorised administrators.
The administrators are expected to be trustworthy and the administration server shall be located in a trusted environment.

<OE.INST> The TOE shall be properly installed. Details for the secure installation options are given in the requirement <R.INST> in section 6.4 of this document.

<OE.PASSW> Unauthorised individuals <S.UNAU> shall not get the password or PIN of an authorised individual <S.AUTH> (any individual knowing any password of the current installation).

Solution:
The users are instructed to keep their password or PIN secret and not to write them down, neither manually nor electronically.
The PC and its environment shall be protected against installing any software programs or hardware devices which enable capturing user password inputs on the keyboard.

<OE.DIRECT> Software which does not use the respective Application Programming Interface of the OS platform for disk access shall not be placed on the PC's hard disk or executed while the computer is operated.

Solution:
The users are instructed not to install or use utility programs like partition managers or disk copy programs while the TOE is installed and active.

<OE.USER> Authorised users shall not actively or negligently compromise the security of the computer on which the TOE is installed.

Solution:
The users are instructed not to install any software which might contain malicious code (like viruses or Trojan horses), not to use any software manipulating the hard disk directly (circumventing the transparent encryption interface), not to modify the TOE program or data files, and not to leave a computer secured by the TOE unattended while being in operational state.

<OE.PHY_CTL> The computer secured by the TOE shall not fall under temporary and undetected physical control of an attacker. If a token or a smart card is

used for authentication, this requirement extends to the token or the smart card and smart card reader.

Solution:
Appropriate physical security measures and physical security policies are in place to manage the risk of this event occurring.

<OE.TOKEN> If tokens or smart cards are used for authentication, these shall have certain properties. Namely, the token or smart card shall:

- implement a secure key storage space for the user's private key in hardware using a secure firmware (card operating system). The firmware shall require a successful PIN authentication before any operation using the private key can be performed and shall not permit reading of the private key even after such an authentication.
- enter a blocked state after at most 5 failed PIN entry attempts, preventing any further PIN entry or data access attempts
- support RSA operations with at least 2048 bits key length on-chip.

If the key pair is generated on the token or imported from outside the TOE (e.g. from a PKI) the operator shall verify that the keys are generated using at least a class K3 RNG as defined in [AIS20].

Solution:
The TOE supports a variety of tokens and smart cards that have the desired properties. The TOE will automatically use cryptographic capabilities of the token if available. The administrator shall confirm that the intended token or smart card type has the required capabilities, and shall ensure the TOE is properly configured to use this token or smart card type.

5.3 TOE Security Policy

Because the TOE security policy is rather simple it is not described in a separate document but added to the Security Target.

The TOE controls access to block devices (hard disk partitions, floppy disks, USB memory sticks, memory cards, compact flash etc.). Each block device is treated as a whole, i.e. there is no specific access control to any subset of data (directories, files) on a block device.

For each user known to the TOE and each block device under control of the TOE it can be defined if the user has access to the block device or not.

As a consequence, one user may have access to several block devices and a set of users may have access to a single block device. Any user not known to the TOE won't have any access to controlled block devices.

The identity and authorization of a user are checked during the boot process of a PC with the TOE installed.

To assert access control the controlled block devices are held completely encrypted. Access to a device is only possible when the TOE assigns the appropriate key to an authenticated user.

The access control to a block device comes into effect when the device is first known to the TOE and when the initial encryption of the device has been completed.

6 IT Security Requirements

The chapter *IT Security Requirements* is divided into the following sections:

TOE Security Functional Requirements - describes the functional requirements for the TOE on basis of CC functional components.

TOE Assurance Requirements – describes the requirements to assure that the TOE implements the functional requirements

Security Requirements for the IT Environment – describes the requirements defined for the IT environment.

Security Requirements for the IT Environment – describes the requirements defined for the non-IT environment.

6.1 TOE Security Functional Requirements

The *TOE Security Functional Requirements* are described using components taken from Part 2 of the Common Criteria.

The listed dependencies for each functional requirement may in some cases not include all options (where options are available on the Common Criteria using the “or” clause), but may list only those dependencies, which are implemented as security functional requirements in the TOE.

Iterations of components are indicated by an additional title added to the name of the SFR, which indicates the specific context of the iterated component.

6.1.1 Class FCS: Cryptographic Support

6.1.1.1 Cryptographic key generation (FCS_CKM.1)

FCS_CKM.1.1
The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm *Extended Random Key Generator* and specified cryptographic key sizes *128 bits (used by AES-128)* and *256 bits (used by AES-256)* that meet the following: *required key length as defined in the standards referred to from each cryptographic algorithm (as shown in sections 6.1.1.3 to 6.1.1.5 of this document)*.

Hierarchical to: no other components.

Dependencies: FCS_COP.1, FCS_CKM.4, FMT_MSA.2

6.1.1.2 Cryptographic key destruction (FCS_CKM.4)

FCS_CKM.4.1
The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key

destruction method *overwriting keys with standard pattern* that meets the following: *no defined standards*.

Hierarchical to: no other components.

Dependencies: FCS_CKM.1, FMT_MSA.2

6.1.1.3 Cryptographic operation (FCS_COP.1)(Device encryption 128)

FCS_COP.1.1
The TSF shall perform *symmetric data encryption and decryption of user data on the block device* in accordance with a specified cryptographic algorithm *AES-128 with CBC mode of operation and block size 128 bits* and cryptographic key size *128 bits* that meet the following: *AES standard as specified in FIPS-197 and CBC mode as specified in NIST Special Publication SP800-38a*.
Refinement: The cryptographic operation is permanently working in the background with each hard disk sector read or write access.

Hierarchical to: no other components.

Dependencies: FCS_CKM.1, FCS_CKM.4, FMT_MSA.2

6.1.1.4 Cryptographic operation (FCS_COP.1)(Device encryption 256)

FCS_COP.1.1
The TSF shall perform *symmetric data encryption and decryption of user data on the block device* in accordance with a specified cryptographic algorithm *AES-256 with CBC mode of operation* and block size *128 bits* and cryptographic key size *256 bits* that meet the following: *AES standard as specified in FIPS-197 and CBC mode as specified in NIST Special Publication SP800-38a*.
Refinement: The cryptographic operation is permanently working in the background with each hard disk sector read or write access.

Hierarchical to: no other components.

Dependencies: FCS_CKM.1, FCS_CKM.4, FMT_MSA.2

6.1.1.5 Cryptographic operation (FCS_COP.1)(Key encryption)

FCS_COP.1.1
The TSF shall perform *symmetric data encryption and decryption of TSF data on the block device* in accordance with a specified cryptographic algorithm *AES-256 in Key Wrap mode* with block size *128 bits* and cryptographic key size *256 bits* that meet the following: *AES standard as specified in FIPS-197 and key wrap mode as specified in RFC 3394*.
Refinement: The cryptographic operation is permanently working in the background with each hard disk sector read or write access.

Hierarchical to: no other components.

Dependencies: [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4, FMT_MSA.2

6.1.1.6 Cryptographic operation (FCS_COP.1)(RSA operation)

FCS_COP.1.1
The TSF shall perform *asymmetric data encryption and decryption of TSF data* in accordance with a specified cryptographic algorithm *RSA* and cryptographic key size *1024, 1536, 2048 or 4096 bits* that meet the following: *PKCS #1 v2.1: RSAES-PKCS1-v1_5 with CRT option used in RSADP.*

Hierarchical to: no other components.

Dependencies: [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4, FMT_MSA.2

6.1.1.7 Cryptographic operation (FCS_COP.1)(Key extraction)

FCS_COP.1.1
The TSF shall perform *user private key extraction* in accordance with a specified cryptographic algorithm *PKCS #12* and cryptographic key size *168 bits* that meet the following: *PKCS #12 using SHA-1 as pseudorandom function and 3-key-Triple-DES as encryption function (Identifier: pbeWithSHAAnd3-KeyTripleDES-CBC, OID: 1.2.840.113549.1.12.1.3); for TripleDES – FIPS 46-3: TDEA keying option 1, used in CBC mode as specified in NIST Special Publication SP800-38a; for SHA1: FIPS 180-3.*

Refinement: The three DES keys are created from the user password according to the PKCS #12 key derivation function, splitting the result of that function into three keys.

Hierarchical to: no other components.

Dependencies: [FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1], FCS_CKM.4, FMT_MSA.2

6.1.1.8 Generation of random numbers (FCS_RND.1)

FCS_RND.1.1
The TSFs shall provide a mechanism for generating random numbers that meet *the functionality class K3 according to [AIS20]*.

FCS_RND.1.2
The TSFs shall be able to enforce the use of TSF-generated random numbers for *key generation.*

Hierarchical to: no other components.

Dependencies: FPT_TST.1

Note 1: This requirement is not contained in CC Part 2 [CC2], but extended. The definition is taken from [AIS31] (BSI, September 2001), and is included as annex to this document.

Note 2: The dependency FPT_TST.1 is intended for true random number generators (TRNG) in [AIS31]. Since the TOE implements a deterministic random number generator and the seed sources are outside the TOE, this functional requirement is not required here.

6.1.2 Class FDP: Access Control Policy

6.1.2.1 Subset Access Control (FDP_ACC.1)

FDP_ACC.1.1
The TSF shall enforce the *SGN device access policy* on *user data on encrypted block devices*.

Hierarchical to: no other components.

Dependencies: FDP_ACF.1

6.1.2.2 Security Attribute Based Access Control (FDP_ACF.1)

FDP_ACF.1.1
The TSF shall enforce the *SGN device access policy* to objects based on the following: *object: user data on block device object – security attribute: key ring of the authenticated user.*

FDP_ACF.1.2
The TSF shall enforce the following rules to determine, if an operation among controlled subjects and controlled objects is allowed: *the device encryption key of a block device is contained in the key ring of the authenticated user*.

FDP_ACF.1.3
The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: *no additional rules*.

FDP_ACF.1.4
The TSF shall explicitly deny access of subjects to objects based on the following additional rules: *no additional rules*.

Hierarchical to: no other components.

Dependencies: FDP_ACC.1, FMT_MSA.3

6.1.3 Class FIA: Identification and Authentication

6.1.3.1 Verification of Secrets (FIA_SOS.1)

FIA_SOS.1.1
The TSF shall provide a mechanism to verify that secrets meet *the password definition rules.*

Refinement: The emergency temporary password change available in the POA is not a password change in the sense of this SFR; the temporary password is not checked against the password rules.

Hierarchical to: no other components.

Dependencies: no dependencies

6.1.3.2 User identification before any action (FIA_UID.2)

FIA_UID.2.1
The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.

Hierarchical to: FIA_UID.1.

Dependencies: no dependencies

6.1.3.3 User authentication before any action (FIA_UAU.2)

FIA_UAU.2.1
The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Refinement: After a user (not the administration server) is successfully authenticated then – and only then – the correct device encryption key for the substantial access to the user data on encrypted devices is provided (with the help of the user's key ring).

Hierarchical to: FIA_UAU.1.

Dependencies: FIA_UID.1

6.1.4 Class FMT: Security management

6.1.4.1 Specification of management functions (FMT_SMF.1)

FMT_SMF.1.1
The TSF shall be capable of performing the following security management functions:
(i) *receive TSF data from administration server*
(ii) *change of password*

Hierarchical to: no other components.

Dependencies: no dependencies

6.1.4.2 Security roles (FMT_SMR.1)

FMT_SMR.1.1
The TSF shall maintain the roles <S.AUTH>.

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

Hierarchical to: no other components.

Dependencies: FIA_UID.1

6.1.4.3 Management of security functions behaviour (FMT_MOF.1)

FMT_MOF.1.1
The TSF shall restrict the ability to *disable* the functions *all security functions* to *authorised*

users.
Refinement: Disable all security functions here means uninstalling the TOE.

Hierarchical to: no other components.

Dependencies: FMT_SMR.1, FMT_SMF.1

6.1.4.4 Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1
The TSF shall restrict the ability to *modify* the *user password* to <*S.AUTH*>.

Hierarchical to: no other components.

Dependencies: FMT_SMR.1, FMT_SMF.1

6.1.5 Class FPT: Protection of the TSF

6.1.5.1 Confidentiality of exported TSF data (FPT_ITC.1)

FPT_ITC.1.1
The TSF shall protect all TSF data transmitted from the TSF to a remote trusted IT product from unauthorised disclosure during transmission.
Refinement: The TSF shall support the protection of cryptographic key material imported from a remote trusted IT product to prevent from unauthorised disclosure during transmission.

Hierarchical to: no other components.

Dependencies: no dependencies

6.2 TOE Assurance Requirements

The *TOE security assurance requirements* are those defined by the *Evaluation Assurance Level 4 (EAL4)*. The requirements are in detail:

Configuration management (Class ACM)
- Partial CM automation (ACM_AUT.1)
- Generation support and acceptance procedures (Component ACM_CAP.4)
- Problem tracking CM coverage (Component ACM_SCP.2)

Delivery and operation (Class ADO)
- Detection of modification (ADO_DEL.2)
- Installation, generation, and start-up procedures (Component ADO_IGS.1)

Development (Class ADV)
- Informal functional specification (Component ADV_FSP.2)
- Security enforcing high-level design (Component ADV_HLD.2)
- Subset of the implementation of the TSF (Component ADV_IMP.1)
- Descriptive low-level design (Component ADV_LLD.1)
- Informal correspondence demonstration (Component ADV_RCR.1)
- Informal TOE security policy model (Component ADV_SPM.1)

Guidance documents (Class AGD)
Administrator guidance (Component AGD_ADM.1)
User guidance (Component AGD_USR.1)

Life cycle support (Class ALC)
Identification of security measures (ALC_DVS.1)
Developer defined life-cycle model (ALC_LCD.1)
Well-defined development tools (ALC_TAT.1)

Tests (Class ATE)
Analysis of coverage (ATE_COV.2)
Testing: high-level design (ATE_DPT.1)
Functional testing (ATE_FUN.1)
Independent testing – sample (Component ATE_IND.2)

Vulnerability assessment (Class AVA)
Validation of analysis (AVA_MSU.2)
Strength of TOE security function evaluation (AVA_SOF.1)
Independent vulnerability analysis (AVA_VLA.2)

The *assurance requirements* are to give evidence that the security functions of the TOE work correctly.

6.3 Security Requirements for the IT Environment

The following *Security Requirements for the IT Environment* are defined:

6.3.1 Class FMT: Security management

Note: The role “administrator” used in the following SFRs for the TOE environment is not defined in section 4.1.1, because this role is not recognised by the TOE directly. However, the administration server, for which these SFRs hold, needs to be able to recognise this role.

6.3.1.1 Management of Security Attributes (FMT_MSA.1)

FMT_MSA.1.1
The TSF shall enforce the *SGN device access policy* to restrict the ability to *create, query, modify and delete* the security attributes *user name, password, key ring* to *administrators*.

Hierarchical to: no other components.

Dependencies: FDP_ACC.1, FMT_SMR.1, FMT_SMF.1

6.3.1.2 Secure Security Attributes (FMT_MSA.2)

FMT_MSA.2.1
The TSF shall ensure that only secure values are accepted for security attributes.

Hierarchical to: no other components.

Dependencies: FDP_ACC.1, FMT_MSA.1, FMT_SMR.1, ADV_SPM.1

6.3.1.3 Static Attribute Initialisation (FMT_MSA.3)

FMT_MSA.3.1
The TSF shall enforce the *SGN device access policy* to provide *permissive* default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2
The TSF shall allow *the administrator* to specify alternative initial values to override the default values when an object or information is created.

Hierarchical to: no other components.

Dependencies: FMT_MSA.1, FMT_SMR.1

6.3.1.4 Management of TSF data (FMT_MTD.1)

FMT_MTD.1.1
The TSF shall restrict the ability to *modify* the *TSF data (hard disk encryption state, device encryption key, boot option)* to *administrators*.

Hierarchical to: no other components.

Dependencies: FMT_SMR.1, FMT_SMF.1

6.4 Security Requirements for the Non-IT Environment

There are the following *Security Requirements for the Non-IT Environment*:

<R.INST> The TOE shall be installed observing the following requirements:

- Installation according to [INSTALL].
- Working network connection of the TOE to an administration server after installation. The TOE and the administration server communicate using a secure SSL connection that is configured as described in the manual for certification compliant operation [CERTCOM].
- Correct preparation of client with client configuration package (as described in the administrator guidance). If smart cards or token are to be used for authentication, the configuration package shall include a policy activating the correct smart card or token support routines.
- Setting of secure attributes in administration and configuration data of the administration database (not part of the TOE) as described in [CERTCOM], such as:
  The minimum length for all passwords must be set to 10 characters.
- If tokens or smart cards are used, these must be properly issued (personalized) for the prospective users and transferred to them in a secure manner.

<R.NOSHAR> No partitions/drives/volumes, directories or files on the local hard disk of the PC secured by the TOE shall be shared with other users when the

PC is connected to a network, in order to avoid placing untrusted software onto the secured PC by using those network shares.

<R.CONN> To update security rules, administration and configuration data, the PC shall be connected to the administration server from time to time.

Note: This requirement <R.CONN> is not strictly implied by one of the security objectives; however it is recommended because it supports all security objectives indirectly.

<R.CAPTKEY> The PC on which the TOE is installed, and the environment where the PC is operated by any authorised user, has to be secured against programs on the PC and against devices which are capable of recording the password entered by an authorised user. Such devices may be keyboard grabbers in the cable between keyboard and PC, which are able to record the keystrokes as well as video cameras capturing the user during password entry. If tokens or smart cards are used for authentication, the connection between token or smart card reader and the PC must be secured against insertion of monitoring devices (e.g. USB capturing device) as well.

<R.DIRECT> The users of the TOE are instructed not to install or run application software which does not use the respective Application Programming Interface of the OS platform for disk access.

<R.PASSW> The users of the TOE are instructed to keep their passwords or PINs secret and not to write them down, neither manually nor electronically.

<R.USER> The users of the TOE are instructed

- not to install any software which might contain malicious code (e. g. viruses or Trojan horses)
- not to use any software manipulating the hard disk directly (circumventing the transparent encryption interface)
- not to modify the TOE program or data files
- not to leave a computer secured by the TOE unattended while being in operational state
- to take their token or smart card with them after powering down the computer or to deposit it in a secure space separate from the computer

<R.SERVER> The administrators using the administration server are expected to be trustworthy and the administration server shall be located in a trusted environment.

<R.PHY_CTL> The computer secured by the TOE shall not fall under temporary and undetected physical control of an attacker. If a token or a smart card is used for authentication, the requirement includes the token or the smart card and smart card reader.

<R.TOKEN> If tokens or smart cards are to be used for authentication, the administrators are expected to select a secure token or smart card type as well as a smart card reader compatible with the TOE (if smart cards are used). Tokens, smart cards, and readers are not part of the TOE.

7 TOE Summary Specification

The chapter *TOE Summary Specification* is organised as follows:

TOE Security Functions – specifies and describes the security functions of the TOE,

Assurance Measures – lists the items and documents provided to fulfil the assurance requirements.

7.1 TOE Security Functions

7.1.1 Overview

Any PC with the TOE installed starts with the user identification and authentication, realised in the “Power On Authentication” <SF1> security function. This function assures that only users authorised for the boot device of the PC are logged in.

The block device access is transparently controlled by data protection function <SF2>, which bases on the encryption of data on the devices.

The security management of the TOE is done remotely by an administration server. The security attributes are provided by this server-based administration via <SF3>.

Keys, which are required locally, are generated on base of random numbers by the random number and key generation security function <SF4>.

7.1.2 Power On Authentication (POA) <SF1>

Power On Authentication is a mechanism of the TOE to check the user’s authenticity before the operating system on a PC is booted from its boot device.

POA is not done under the control of the PC operating system, but a small separate operating system is booted first to control the POA function.

With POA installed, depending on the type of authentication, the system prompts (i) for a valid user name and a password or (ii) for a CryptoToken and for a PIN after starting the PC. Note: SGN – Device Encryption provides further authentication mechanisms that are not covered by the evaluation.

For user name and password authentication:

The user name and the password are entered via keyboard, when a login prompt is displayed on the screen. Only a correct combination of user name and password – or the correct password for the defined standard user (if any) – enables to boot the operating system. In case of an incorrect password entered, the POA module waits for some time until the next password entry is possible. This time increases for each incorrect entry.

For CryptoToken authentication:

When CryptoToken is specified as authentication device, the user has to enter the PIN for accessing the CryptoToken, i.e. two factors are necessary to authenticate the user.

If the authentication passes successfully, the TOE activates the keys needed to boot the PC. Then the TOE continues operation with booting the regular operating system.

If the PC's boot device is encrypted, POA includes a mechanism which calculates the device encryption key for the boot device (for the use of the encryption key see <SF2>). The user's key ring is compiled from one or more key tables after a successful logon to Windows.

All users using password authentication are enabled to change their password during the operation of the TOE. For the user, this works like changing the password for the Windows operating system. The quality of a new password is checked by the TOE against a set of password rules (minimum length, password history etc.) before the new password is passed on to Windows and the password change is allowed to proceed. The entered password is automatically synchronized between the Windows operating system and the TOE.

There is one exception to the mechanism described above. If the user chooses to temporarily set a different password inside the POA, this password is not checked against the password policy, nor is it ever synchronized with the Windows password. This option exists as an emergency measure to restore protection in the face of a compromised user password. It solves the problem that a domain user cannot use the regular Windows password change as long as the domain is not accessible (e.g. for a notebook used on a plane). As soon as a regular password change as described in the previous paragraph takes place, the temporary password is replaced by the new one, which is again checked against the password rules.

This function implements the SFRs *Cryptographic operation (FCS_COP.1)(RSA* operation), *Cryptographic operation (FCS_COP.1)(Key* extraction)*, Verification of Secrets (FIA_SOS.1), User identification before any action (FIA_UID.2)*, *User authentication before any action (FIA_UAU.2), Specification of management functions (FMT_SMF.1), Management of TSF data (FMT_MTD.1) and Security roles (FMT_SMR.1)*.

The security function <SF1> is claimed to have the strength of security functions *SOF-medium*.

7.1.3 Protection of Data on Protected Devices <SF2>

After a successful authentication by security function <SF1>, the cryptographic keys needed to boot the PC are determined out of the user's key ring stored in TSF data. The user's key ring is compiled from one or more key tables after a successful logon to Windows. An access to any encrypted device is only possible if the cryptographic key used for encryption of that specific device is known. Hence, the security function <SF2> ensures that data provided by authorised users are protected when being stored on encrypted devices and when the PC is not in operation or the device is detached from a PC in operation.

The used encryption algorithm can be selected from the range of available system algorithms (AES-128 or AES-256 in CBC mode).

The key for the encryption is either generated by a random key generator (see <SF4>) or predefined by the administration over the server-based administration interface.

All write and read accesses to the encrypted devices are intercepted by one of the encryption handlers (INT 13h handler or Windows device driver) depending on the state of the system. On a write access, the data is encrypted; on a read access, the data is decrypted.

Using an encrypted device on a PC which has not been booted with POA and user authentication results in a state where information can't be obtained from the device(s) as a result of the encryption. The same is true for built-in devices when the PC is booted from a floppy disk or any other bootable device and the POA is not run.

When booting a secured PC from hard disk, the control is handed from one part of the TOE to another. First, the POA module checks the authenticity of the user and calculates the device encryption key for the boot device (normally a hard disk partition). Next, an INT 13h handler is installed to decrypt the hard disk data during the boot phase. This handler remains active as long as hard disk access is performed during BIOS INT 13h (DOS e.g.). When the Windows operating system is booted a device driver is automatically loaded, which is taking over hard disk on-line encryption and decryption during the Windows session.

Encryption state changes of any device are defined by using the administration server via the remote connection. As a result of such a change initial encryption or complete decryption of a device is automatically invoked by the TOE. If the initial encryption or complete decryption is interrupted (e.g. by shutdown), it continues automatically after booting the PC again. This encryption task is performed by a background process, which is started after POA, but before the operating system user logon.

This function implements the SFRs Cryptographic operation (FCS_COP.1)(Device encryption 128), Cryptographic operation (FCS_COP.1)(Device encryption 256), *Subset Access Control (FDP_ACC.1), Security Attribute Based Access Control (FDP_ACF.1)*

The security function <SF2> is not claimed a strength of security functions.

7.1.4 Secure Server-Based Administration <SF3>

The administration of the TOE is done in the administration server. The TOE retrieves its administration data from the administration server over a network connection.

Besides the TOE installation, uninstallation and user password change, there is no administration function available at the client side for the TOE.

The local administration data (TSF data) is secured by symmetric encryption. Only a successful identification and authentication grants access to the TSF data.

The connection to the administration server is performed over a web server connection. The administration data is transmitted from the administration server to the TOE using this connection. Sensitive administration data is protected during transport. For example cryptographic keys are encrypted using the AES-256 key wrap algorithm. If new administration data is received concerning built-in or attached devices, the resulting actions (e.g. initial device encryption or final decryption) are immediately invoked.

Note: The TOE and the administration server communicate using a secure SSL connection that is provided by the environment. All data transmitted over this secure connection is encrypted and its integrity is protected. Thus, sensitive administration data is protected by multiple protection layers during transmission.

This function implements the SFRs *Cryptographic key destruction (FCS_CKM.4)*, *Cryptographic operation (FCS_COP.1)(Key* encryption*), Specification of management*

functions (FMT_SMF.1), *Management of security functions behaviour (FMT_MOF.1)* and *Confidentiality of exported TSF data (FPT_ITC.1)*.

The security function <SF3> is not claimed a strength of security functions.

7.1.5 Random Number Generation and Key Generation <SF4>

During installation of the TOE and initial encryption of local block devices a deterministic random number generator (DRNG) is used for the generation of the cryptographic keys. This applies to the following keys:

- Machine dependent key encryption key
- Device encryption keys of all local block devices

The DRNG fulfils the requirements of class K3 as described in [AIS20]. A detailed description is provided in a separate document.

This function implements the SFRs *Cryptographic key generation (FCS_CKM.1), Generation of random numbers (FCS_RND.1).*

The random number generation (DRNG) within <SF4> is claimed to have the strength of security functions *SOF-medium*.

The key generation part of <SF4> is not claimed a strength of security functions.

7.1.6 Further Functions of SafeGuard Enterprise – Device Encryption (informative only)

SafeGuard Enterprise – Device Encryption supports some more functions for the convenience of secure operation and administration of PCs. **The following functions** are included into the product, but **are not part of the evaluated functions of the TOE**.

Auditing

The TOE supports the recording of events (e.g. administration actions, user login) to the Windows event database and to an own audit database. These auditing functions are not part of this evaluation.

Challenge-Response Login

During POA a Challenge/Response Logon is possible using a challenge-response procedure. For this function, the POA module generates a random challenge string, which can be transmitted by the user to a security officer with access to the SafeGuard Management Center. With the Response Generation Program, the remote user creates a response string out of the challenge, his password and a function code. Then the user enters this response string at the POA and is then enabled to perform the functions the remote user has permitted. The response code is only valid for a single login within a limited time span. The function of challenge-response login is not a part of this evaluation.

Local Self Help

This function can be enabled and used for situations where users cannot log in at POA and a help desk for Challenge/Response is not available. Users can regain access to the system by answering a defined number of randomly selected security questions. The function needs

to be enabled and deployed by the administrator and will only become active after a user has answered enough questions under Windows. This functionality is not part of this evaluation.

Service Accounts

These are deployed in the form of a simple service account list. Such a list simply enumerates specific Windows accounts that need not become permanently assigned to a system (meaning they cannot be used to log into the system at POA). A maintenance account is an obvious candidate for this list – it must be usable on many machines but must not be usable to boot all these machines in the absence of their owners. The feature is not part of the evaluation; however, it has no impact on any of the TSF.

POA Access Accounts

These are actual accounts that need to be assigned to machines and deployed as such. Contrary to the accounts on the Service Account List, these accounts work only in POA and don't necessarily even exist in Windows. They can be used as maintenance accounts as well, e.g. to boot a system to the Windows logon so that it can connect to the network for provisioning. If the user only holds a POA Access Account, this is basically all they can do with the system. This feature is not part of this evaluation; it also has no impact on any TSF.

Fast Initial Encryption

This is an improved initial encryption mode that can save significant time by skipping over unused areas of partitions. However, this can only be used without security implications if the system's hard disk has not held any sensitive information prior to encryption. In this evaluation we assume this feature is not used.

Bitlocker Support

On Windows 7 systems, the disk encryption client can also use Bitlocker and Bitlocker-To-Go encryption instead of its built-in sector-based encryption. In this mode, the system will use Bitlocker Power-On-Authentication instead of the native SafeGuard POA. As a result, mixed environments using both Bitlocker and SafeGuard encryption can be managed from one central SafeGuard console. However, Bitlocker support and management are not part of this evaluation, i.e. clients must be configured to prefer SafeGuard encryption over Bitlocker.

OPAL Self-Encrypting Drive Support

Similar to managing Bitlocker encryption, the SafeGuard Client can also be used to manage an OPAL drive if available. The Power-On Authentication remains the same, instead of software encryption the hardware encryption of the drive is used. However, this implies that all partitions of the drive use the same key (the drive password). OPAL mode is not part of this evaluation, as the drive would need to be evaluated as well.

Self Tests

During start-up of the PC and the TOE's security functions, a self test mechanism assures the integrity of the major parts of the TOE. These self tests consist of:

(i) An integrity test of the SafeGuard Enterprise – Device Encryption kernel containing the security mechanisms for the TSF data, the POA code and the symmetric encryption algorithms used during POA,

(ii) A known-answer test of the cryptographic algorithms used for POA,

(iii) A known-answer test of the symmetric encryption algorithms for the protected mode operation phase.

These mechanisms are not part of this evaluation.

SafeGuard Data Exchange

This feature is included in the basic encryption client. It can be used to encrypt individual files for data exchange with other parties inside or outside the corporation. Inside the corporation, the user can select a key from the key ring that the recipient also has access to. For exchange with outside parties, the user can generate a local key from an arbitrary password. The password can then be given to the external party. If the external party does not have an SGN client, a special application, *SafeGuard Portable*, can be used to access the encrypted files. While the data exchange feature uses the same file-based encryption mechanism as the client, this functionality is not part of the evaluation.

SafeGuard Enterprise standalone

SafeGuard Enterprise can also be operated in standalone mode. SafeGuard Offline Clients in standalone mode can be managed by creating policies in the SafeGuard Policy Editor and distributing them via third party mechanisms to the SafeGuard Offline Clients. This scenario is suitable for smaller enterprise environments.

Configuration Protection

This feature is included on the TOE media but resides in a separate installation file. It offers functions like port control and data leakage prevention. However, these mechanisms are not part of the TOE and not relevant for this evaluation.

7.2 Assurance Measures

Appropriate documentation will be provided to satisfy the Security Assurance Requirements described in section 6.2.

The TOE itself does not provide any measure or mechanism to satisfy the assurance requirements. Assurance is guaranteed by the development process and by the users observing the corresponding directions.

The following table associates the measures and the documents describing them with the assurance requirements of CC EAL4:

CC Requirement	Assurance Measure	Describing Document(s)
ACM_AUT.1	The developers use a configuration management system, which enables only authorised changes to the configuration items. This CM system is partially automated. The configuration management system and its functionality is documented.	Configuration Management and Development Security Documentation, Utimaco
ACM_CAP.4	The developers use a configuration management system, which allows to uniquely identify all configuration items and supports the generation of the TOE. The effectiveness of the CM system is made evident. A configuration list is provided.	Configuration Management and Development Security Documentation, Utimaco SafeGuard Enterprise – Device Encryption V5.60: Configuration List

CC Requirement	Assurance Measure	Describing Document(s)
ACM_SCP.2	The coverage of all configuration items identified for the TOE by the configuration management is documented in the configuration management documentation.	Configuration Management and Development Security Documentation, Utimaco SafeGuard Enterprise – Device Encryption V5.60: Configuration List, Utimaco
ADO_DEL.2	The delivery procedures for the TOE are described in a separate document.	Secure Delivery Documentation, Utimaco
ADO_IGS.1	The installation, generation and start-up of the TOE are described in a separate document.	SafeGuard Enterprise – Device Encryption V5.60: Installation, Generation and Start-Up, Utimaco
ADV_FSP.2	The informal functional specification describing the external interfaces is specified in a separate document.	SafeGuard Enterprise – Device Encryption V5.60: Functional Specification and Correspondence Demonstration, Utimaco
ADV_HLD.2	The security enforcing high-level design is provided in a separate document.	SafeGuard Enterprise – Device Encryption V5.60: High Level Design and Correspondence Demonstration, Utimaco
ADV_IMP.1	The TSF implementation is pure software. It is accessible through the Utimaco configuration management system. The assignment between TSF and source code files is described in the low-level design documentation.	SafeGuard Enterprise – Device Encryption V5.60: Set of code module documentation Files generated from source code, Utimaco
ADV_LLD.1	The descriptive low-level design is provided in a separate document.	SafeGuard Enterprise – Device Encryption V5.60: Low Level Design and Correspondence Demonstration, Utimaco SafeGuard Enterprise – Device Encryption V5.60: Set of code module documentation Files generated from source code, Utimaco
ADV_RCR.1	The correspondence demonstration is explained in the documents together with the functional specification, the high-level design and the low-level design	SafeGuard Enterprise – Device Encryption V5.60: Functional Specification and Correspondence Demonstration, Utimaco SafeGuard Enterprise – Device Encryption V5.60: High Level Design and Correspondence Demonstration, Utimaco SafeGuard Enterprise – Device Encryption V5.60: Low Level Design and Correspondence Demonstration, Utimaco

CC Requirement	Assurance Measure	Describing Document(s)
ADV_SPM.1	The informal security policy model is contained in section 5.3 of this document.	
AGD_ADM.1		SafeGuard Enterprise Version 5.60 - Administrator Help SafeGuard Enterprise Version 5.60 – Installation Manual
AGD_USR.1		SafeGuard Enterprise Version 5.60 - User Help SafeGuard Enterprise Version 5.60 - Installation Manual
ALC_DVS.1	The measures taken to assure the security during the development of the TOE are described in a separate document together with the configuration management system.	Configuration Management and Development Security Documentation, Utimaco
ALC_LCD.1	The life cycle model is described in a separate document together with the configuration management system.	Configuration Management and Development Security Documentation, Utimaco
ALC_TAT.1	The development tools used for the construction of the TOE are defined in a separate document together with the configuration management system.	Configuration Management and Development Security Documentation, Utimaco
ATE_COV.2	The analysis of the test coverage is provided in a separate document.	SafeGuard Enterprise – Device Encryption V5.60: Test Description and Analysis, Utimaco
ATE_DPT.1	The analysis of the depth of testing in accordance with the high-level design is described in a separate document.	SafeGuard Enterprise – Device Encryption V5.60 Test Description and Analysis, Utimaco
ATE_FUN.1	The functional testing is described in a separate document; the test cases are described in the product's test plan.	SafeGuard Enterprise – Device Encryption V5.60: Test Description and Analysis, Utimaco SafeGuard Enterprise V5.60: Test Plan, Utimaco
ATE_IND.2	Independent Testing is carried out by the evaluation facility.	-
AVA_MSU.2	The evaluation of the guidance documents are provided in a separate document.	SafeGuard Enterprise – Device Encryption V5.60: Vulnerability Assessment, Utimaco
AVA_SOF.1	The evaluation of the strength of security functions is provided in a separate document.	SafeGuard Enterprise – Device Encryption V5.60: Vulnerability Assessment, Utimaco

CC Requirement	Assurance Measure	Describing Document(s)
AVA_VLA.2	A vulnerability assessment by the developer is provided in a separate document. Independent vulnerability analysis and penetration tests are to be performed by the evaluator.	SafeGuard Enterprise – Device Encryption V5.60: Vulnerability Assessment, Utimaco

8 PP Claims

This Security Target does not make any claim that the TOE is conformant with the requirements of a *Protection Profile*. As a consequence, the sections "*PP Reference*", "*PP Refinement*" and "*PP Additions*" are omitted.

9 Rationale

The chapter *Rationale* is divided into the following sections:

Security Objectives Rationale – describing the relations between threats and security objectives,

Security Requirements Rationale – describing the relations between security objectives and security requirements res. security assurance requirement,

Dependency Rationale – describing the support of dependencies among the requirements.

TOE Summary Specification Rationale – describing the relations between the security requirements and the TOE's security functions and between the assurance requirements and the assurance measures,

PP Claims Rationale – describing the relations to a claimed Protection Profile.

The purpose of the ST rationale is to demonstrate that a complete, coherent and internally consistent set of security objectives, security requirements, IT security functions and assurance measures have been proposed to satisfy the identified security problem.

9.1 Security Objectives Rationale

It shall be demonstrated that the *Security Objectives* (chapter 5) are appropriate referring to the aspects of the *Security Environment* (chapter 4).

The stated *Security Objectives* (chapter 5) address all of the identified *Secure Usage Assumptions* (section 4.2) and *Threats* (section 4.3).

The following table shows that each security objective addresses at least one threat or assumption:

Objective	**Threats, Assumptions**
<O.ACCESS>	<T.ACCESS>
<O.MANAGE>	<T.REMADMIN>
<O.KEYGEN>	<T.KEYGEN>
<OE.SERVER>	<T.REMADMIN>, <A.ADMIN>
<OE.DIRECT>	<A.DIRECT>
<OE.PASSW>	<A.PASSW>
<OE.INST>	<A.INST>
<OE.USER>	<A.USER>
<OE.PHY_CTL>	<A.PHY_CTL>
<OE.TOKEN>	<A.TOKEN>

The table shows that each *Security Objective* maps to at least one *Threat* or *Assumption*.

Each threat or assumption is covered by at least one security objective.

Detailed Explanation / Justification:

<O.ACCESS> prevents unauthorised individuals <S.UNAU> from substantial access <ACC.SUB> to user data stored on the hard disk partitions <D.USER> after the PC has been switched off or if booted with an operating system different from Windows. This security objective exactly counters threat <T.ACCESS>.

<O.MANAGE> prevents unauthorised individuals <S.UNAU> from performing management operations on the TOE. This covers the manipulation of administration data received from the remote administration server and local password change. So this objective counters the threat <T.REMADMIN>.

<OE.SERVER> guarantees that only authorised individuals provide configuration data for the TOE on the administration server. So this objective also counters the threat <T.REMADMIN>. In addition it supports the assumption <A.ADMIN> that administrators are trustworthy and that the administration server is located securely.

<O.KEYGEN> forces generation of secure and unpredictable cryptographic keys and thus counters the threat <T.KEYGEN>.

<OE.DIRECT> claims that software which does not use the respective Application Programming Interface of the OS platform, is not installed or executed while the TOE is installed on the PC. This security objective exactly covers the assumption <A.DIRECT>.

<OE.PASSW> claims that no unauthorised individual <S.UNAU> will be able to get knowledge of a password of any authorised individual <S.AUTH>. This security objective exactly covers the assumption <A.PASSW>.

<OE.INST> claims that the TOE is installed and configured with proper installation options and security attributes. This security objective exactly covers the assumption <A.INST>.

<OE.USER> claims that authorised users of the TOE do not compromise the security of the TOE actively or negligently. This includes introduction of malicious software or manipulation of harddisk, TOE or TSF-data. This exactly covers the assumption <A.USER>.

<OE.PHY_CTL> claims that additional physical security measures and physical security policies are in place to prevent that attackers gain temporary and undetected access to computers secured by the TOE while being in operational state. If a token or smart card is used for user authentication, the claim includes the token or smart card and smart card reader as well to prevent manipulation of these devices. This directly covers <A.PHY_CTL>.

<OE.TOKEN> claims the minimum security capabilities of the token or smart card used for authentication – if such a device is used. This covers the assumption <A.TOKEN>.

9.2 Security Requirements Rationale

It shall be demonstrated that the set of *Security Requirements* (TOE and environment, chapter 6) is suitable to meet and traceable to the *Security Objectives* (chapter 5).

9.2.1 Security Functional Requirements

The *TOE Functional Requirements* (section 6.1) and the *Security Requirements for the IT Environment* (section 6.3) can be mapped to the *TOE Security Objectives* (section 5.1) as follows:

SFR	<O.ACCESS>	<O.MANAGE>	<O.KEYGEN>
TOE SFR			
FCS_CKM.1	X	X	
FCS_CKM.4	X	X	
FCS_COP.1 (Device encryption 128)	X		
FCS_COP.1 (Device encryption 256)	X		
FCS_COP.1 (Key encryption)		X	
FCS_COP.1 (RSA operation)	X		
FCS_COP.1 (Key extraction)	X		
FCS_RND.1			X
FDP_ACC.1	X		
FDP_ACF.1	X		
FIA_SOS.1		X	
FIA_UID.2	X	X	
FIA_UAU.2	X	X	
FMT_SMF.1		X	
FMT_SMR.1	X	X	
FMT_MOF.1		X	
FMT_MTD.1		X	
FPT_ITC.1		X	

The table shows that each TOE Security Objective is implemented by more than one SFR. The table also shows that each SFR addresses at least one TOE Security Objective.

Detailed Explanation / Justification:

The goal to protect user data on block devices <O.ACCESS> is reached with a combination of user authentication and data encryption. Access control is provided by *Subset Access Control (FDP_ACC.1)* and *Security Attribute Based Access Control (FDP_ACF.1)*. It is implemented by the means of cryptographic operations *Cryptographic operation (FCS_COP.1)(Device encryption 128)* to *(Device encryption 256)*. The cryptographic operations are supplied with keys generated by *Cryptographic key generation (FCS_CKM.1)* and destroyed by *Cryptographic key destruction (FCS_CKM.4)*.

All authorised users are assigned the role *authorised user*, which is defined by the SFR *Security roles (FMT_SMR.1)*. Authorised users are determined with the help of the SFRs *User identification before any action (FIA_UID.2)* and *User authentication before any action*

(FIA_UAU.2). If not using user name/password authentication, the user can use a cryptographic token with a certificate on it, which is processed by *Cryptographic operation (FCS_COP.1)(RSA operation)* and *Cryptographic operation (FCS_COP.1)(Key extraction).*

The goal to protect the TOE management operations <O.MANAGE> (changing the protection status of the TOE or modifying other TSF data) is addressed by the SFRs *Specification of management functions (FMT_SMF.1)* and *Management of security functions behaviour (FMT_MOF.1)*. The change of the password of an authorised user is restricted to this authorised user and to the administration server by the SFR *Management of TSF data (FMT_MTD.1)*.These functions rely on the role *authorised user*, which is defined by the SFR *Security roles (FMT_SMR.1)*. Authorised users are determined with the help of the SFRs *User identification before any action (FIA_UID.2)* and *User authentication before any action (FIA_UAU.2)*.

To secure local TSF data against unauthorised access, the cryptographic operation *Cryptographic operation (FCS_COP.1)(Key* encryption) is used, which is supplied by a key generated by *Cryptographic key generation (FCS_CKM.1)* and destroyed by *Cryptographic key destruction (FCS_CKM.4)*

The change of passwords is checked using the SFR *Verification of Secrets (FIA_SOS.1)*.

The management of access control is done remotely and administration data is transferred to the TOE by an external interface. The path between the remote administration and the TOE is secured by *Confidentiality of exported TSF data (FPT_ITC.1)*

The generation of keys by using high quality random numbers <O.KEYGEN> is realised with the help of *Generation of random numbers (FCS_RND.1)*.

9.2.2 Security Requirements for the Environment

The *Security Objectives for Environment* (section 5.2) are covered by the defined *Security Requirements for the IT Environment* and the *Security Requirements for the Non-IT Environment* (section 6.4), as the references in the following table show.

Objective	**Requirement**
<OE.SERVER>	FMT_MSA.1 FMT_MSA.2 FMT_MSA.3 FMT_MTD.1 <R.SERVER>
<OE.INST>	<R.INST>
<OE.PASSW>	<R.PASSW> and <R.CAPTKEY>
<OE.DIRECT>	<R.DIRECT>
<OE.USER>	<R.USER>, <R.NOSHAR>
<OE.PHY_CTL>	<R.PHY_CTL>
<OE.TOKEN>	<R.TOKEN> <R.INST>

The table shows that each Security objective for the environment is covered by a single requirement or a combination of requirements either for IT environment or for Non-IT environment.

The table also shows that each requirement (except <R.CONN>) is necessary to match the security objectives for the IT environment.

Requirement <R.CONN> is specified to guarantee that the security attributes of the TOE are synchronised with the managed attributes on the remote administration server and thereby supports all security objectives indirectly.

Detailed Explanation / Justification:

To provide secure administration data for the remote administration of the TOE (<OE.SERVER>), the administration server must implement the SFRs *Management of Security Attributes (FMT_MSA.1)*, *Secure Security Attributes (FMT_MSA.2)*, *Static Attribute Initialisation (FMT_MSA.3)* and *Management of TSF data (FMT_MTD.1)*. In addition secure location of the administration server and trusted administrators are needed for this as required by <R.SERVER>.

To install the TOE properly (<OE.INST>), the measures listed in the requirement <R.INST> have to be regarded.

Disclosure of the password or PIN can be performed by three different possibilities:

- An authorised user may write down his password/PIN or may tell his password to an unauthorised individual. This shall be avoided by following the instruction to keep the password/PIN secret as defined in <R.PASSW>.
- An electronic device is inserted by an attacker somewhere between the keyboard and the PC keyboard processor. This device is capable of recording the keystrokes and can be removed later and its memory can be read out. To avoid this the PC must be secured in a way that the insertion of such a device is not possible or that it can be easily detected by the user. This requirement is defined in <R.CAPTKEY>.
- The room where the PC is operated and the user enters the password is monitored by a video camera and the image is recorded. In this way an unauthorised individual could get information about the user's password or PIN. To avoid this the PC environment must be checked that such monitoring is not present or cannot record the user's keyboard entries. This requirement is also defined in <R.CAPTKEY>.

Both requirements together avoid the disclosure of the user's password <OE.PASSW>.

To prevent from the usage of software, which does not use the disk access API of the OS and therefore circumvents the access control function of the TOE (<OE.DIRECT>), the users are instructed not to install such software on the secured PC (<R.DIRECT>).

To make sure that authorised users do not compromise the security of the TOE (<OE.USER>), adequate instructions have to be given, as required by <R.USER>. A special case of compromise would occur if an authorised user would allow access of other (potentially untrusted) users over a network. To prevent this (with the danger of disclosure of passwords or modification of security attributes or security mechanisms by intruding of non-trusted software over network), no network shares shall be defined on the target system. This is assured by the Non-IT requirement <R.NOSHAR> where it is defined, that no user has access via network connections to the local hard disk(s) of the secured PC.

The requirement <R.PHY_CTL> prevents use of a potentially compromised TOE after (potential) access by unauthorised persons. This supports directly the corresponding objective <OE.PHY_CTL>.

If smart cards or tokens are to be used for authentication, <R.TOKEN> mandates selection and deployment of the proper hardware. <R.INST> ensures the token or smart card is properly initialized, and that the client is correctly configured to accept it as an authentication device. Both requirements therefore ensure <OE.TOKEN> is properly implemented.

9.2.3 Assurance Requirements and Strength of Security Functions

The *TOE Security Functional Requirements* (section 6.1) cover all aspects to ensure that the security functions provided by the TOE are actually able to respond to the security problems defined in form of *TOE Security Objectives* (section 5.1). The assurance requirements are those defined for the Evaluation Assurance Level 4. So, there is no need to further demonstrate that these requirements are useful and suitable.

The claimed rating of the minimum strength of security functions for the configuration of the TOE mentioned in section 4.2 (“Secure Usage Assumptions“) is *SOF-medium*. These requirements match with the background of identified threats (section 4.3) on the one hand and to the security environment (refer to section 4.1) on the other. From the requirements of the environment the *Security Objectives* were derived very straightforward (see section 9.1). So, it is argued that the claimed rating of the minimum strength of security functions is also consistent with the *Security Objectives*. All security functions base upon mechanisms to which either a strength can be assigned or which may not be overcome, when implemented correctly. The strength of that underlying mechanisms and with them the strength of security functions is discussed in the document ”SafeGuard Enterprise – Device Encryption V5.60: Vulnerability Assessment, Utimaco, 2011” under the aspect of the assurance requirement AVA_SOF.1.

9.3 Dependency Rationale

9.3.1 Functional Requirements Dependencies

The following tables show all functional requirements dependencies required by the TOE and IT-environment.

[paragraph numbers in parenthesis refer to the appropriate paragraph in this document]

Component	Dependencies	Dependency fulfilled by
TOE security functional components		
FCS_CKM.1	FCS_COP.1 FCS_CKM.4 FMT_MSA.2	FCS_COP.1 (Device encryption 128), FCS_COP.1 (Device encryption 256), partially also FCS_COP.1 (Key encryption),(6.1.1.3, 6.1.1.4, 6.1.1.5,), since the TOE generates keys for these algorithms (not all possible keys in case of FCS_COP.1 (Key encryption)) FCS_CKM.4 (6.1.1.2) FMT_MSA.2 (6.3.1.2) (provided by the environment)
FCS_CKM.4	FCS_CKM.1 FMT_MSA.2	FCS_CKM.1 (6.1.1.1) FMT_MSA.2 (6.3.1.2) (provided by the environment)
FCS_COP.1 (Device encryption 128),	FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	FCS_CKM.1 (6.1.1.1) FCS_CKM.4 (6.1.1.2) FMT_MSA.2 (6.3.1.2) (provided by the environment)
FCS_COP.1 (Device encryption 256),	FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	FCS_CKM.1 (6.1.1.1) FCS_CKM.4 (6.1.1.2) FMT_MSA.2 (6.3.1.2) (provided by the environment)
FCS_COP.1 (Key encryption)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	Some Key encryption keys are generated by the TOE, which is covered by FCS_CKM.1 (6.1.1.1), some Key encryption keys are imported via the administrative interface. The security of this import is covered by FPT_ITC.1, which was refined to also cover data import, (6.1.5.1) thereby replacing FDP_ITC.* FCS_CKM.4 (6.1.1.2) FMT_MSA.2 (6.3.1.2) (provided by the environment)
FCS_COP.1 (RSA operation)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1] FCS_CKM.4 FMT_MSA.2	The asymmetric keys are imported via the administrative interface. The security of this import is covered by FPT_ITC.1, which was refined to also cover data import, (6.1.5.1) thereby replacing FDP_ITC.*. FCS_CKM.4 (6.1.1.2) FMT_MSA.2 (6.3.1.2) (provided by the environment)

FCS_COP.1 (Key extraction)	[FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1]	According to FCS_COP.1 (Key extraction) the keys used for the key extraction are derived from the user password as specified in PKCS #12 (which again refers to PKCS #5), therefore neither key generation nor key import are necessary here.
	FCS_CKM.4	FCS_CKM.4 (6.1.1.2)
	FMT_MSA.2	FMT_MSA.2 (6.3.1.2) (provided by the environment)
FCS_RND.1	FPT_TST.1	See note below
FIA_SOS.1	none	---
FIA_UID.2	none	---
FIA_UAU.2	FIA_UID.1	FIA_UID.2 (6.1.3.2)
FMT_SMF.1	None	---
FMT_SMR.1	FIA_UID.1	FIA_UID.2 (6.1.3.2)
FMT_MOF.1	FMT_SMR.1	FMT_SMR.1 (6.1.4.2)
	FMT_SMF.1	FMT_SMF.1 (6.1.4.1)
FMT_MTD.1	FMT_SMR.1	FMT_SMR.1 (6.1.4.2)
	FMT_SMF.1	FMT_SMF.1 (6.1.4.1)
FPT_ITC.1	None	---
Security Requirements for the IT Environment		
FMT_MSA.1	FDP_ACC.1	FDP_ACC.1 (6.1.2.1)
	FMT_SMR.1	FMT_SMR.1 (6.1.4.2)
	FMT_SMF.1	FMT_SMF.1 (6.1.4.1)
FMT_MSA.2	FDP_ACC.1	FDP_ACC.1 (6.1.2.1)
	FMT_MSA.1	FMT_MSA.1 (6.3.1.1)
	FMT_SMR.1	FMT_SMR.1 (6.1.4.2)
	ADV_SPM.1	Security Policy (5.3)
FMT_MSA.3	FMT_MSA.1	FMT_MSA.1 (6.3.1.1)
	FMT_SMR.1	FMT_SMR.1 (6.1.4.2)
FMT_MTD.1	FMT_SMR.1	FMT_SMR.1 (6.1.4.2)
	FMT_SMF.1	FMT_SMF.1 (6.1.4.1)

Note (regarding dependency of FCS_RND.1 on FPT_TST.1): As already noted in section 6.1.1.8, the dependency FPT_TST.1 is intended for true random number generators (TRNG) in [AIS31]. Since the TOE implements a deterministic random number generator and the seed sources are outside the TOE, this functional requirement is not required here.

9.3.2 Assurance Requirements Dependencies

The following table shows that all assurance requirements dependencies are fulfilled.

Item	Assurance Req.	Dependencies	Dependency fulfilled by
#1	ACM_AUT.1	ACM_CAP.3	ACM_CAP.4 (#2)
#2	ACM_CAP.4	ALC_DVS.1	ALC_DVS.1 (#14)
#3	ACM_SCP.2	ACM_CAP.3	ACM_CAP.4 (#2)
#4	ADO_DEL.2	ACM_CAP.3	ACM_CAP.4 (#2)
#5	ADO_IGS.1	AGD_ADM.1	AGD_ADM.1 (#12)
#6	ADV_FSP.2	ADV_RCR.1	ADV_RCR.1 (#10)
#7	ADV_HLD.2	ADV_FSP.1 ADV_RCR.1	ADV_FSP.2 (#6) ADV_RCR.1 (#10)
#8	ADV_IMP.1	ADV_LLD.1 ADV_RCR.1 ALC_TAT.1	ADV_LLD.1 (#9) ADV_RCR.1 (#10) ALC_TAT.1 (#16)
#9	ADV_LLD.1	ADV_HLD.2 ADV_RCR.1	ADV_HLD.2 (#7) ADV_RCR.1 (10)
#10	ADV_RCR.1	none	---
#11	ADV_SPM.1	ADV_FSP.1	ADV_FSP.2 (#6)
#12	AGD_ADM.1	ADV_FSP.1	ADV_FSP.2 (#6)
#13	AGD_USR.1	ADV_FSP.1	ADV_FSP.2 (#6)
#14	ALC_DVS.1	none	---
#15	ALC_LCD.1	none	---
#16	ALC_TAT.1	ADV_IMP.1	ADV_IMP.1 (#8)
#17	ATE_COV.2	ADV_FSP.1 ATE_FUN.1	ADV_FSP.2 (#6) ATE_FUN.1 (#19)
#18	ATE_DPT.1	ADV_HLD.1 ATE_FUN.1	ADV_HLD.2 (#7) ATE_FUN.1 (#19)
#19	ATE_FUN.1	none	---
#20	ATE_IND.2	ADV_FSP.1 AGD_ADM.1 AGD_USR.1 ATE_FUN.1	ADV_FSP.2 (#6) AGD_ADM.1 (#12) AGD_USR.1 (#13) ATE_FUN.1 (#19)
#21	AVA_MSU.2	ADO_IGS.1 ADV_FSP.1 AGD_ADM.1 AGD_USR.1	ADO_IGS.1 (#5) ADV_FSP.2 (#6) AGD_ADM.1 (#12) AGD_USR.1 (#17)
#22	AVA_SOF.1	ADV_FSP.1 ADV_HLD.1	ADV_FSP.2 (#6) ADV_HLD.2 (#7)

#23	AVA_VLA.2	ADV_FSP.1 ADV_HLD.2 ADV_IMP.1 ADV_LLD.1 AGD_ADM.1 AGD_USR.1	ADV_FSP.2 (#6) ADV_HLD.2 (#7) ADV_IMP.1 (#8) ADV_LLD.1 (#9) AGD_ADM.1 (#12) AGD_USR.1 (#13)

9.4 TOE Summary Specification Rationale

9.4.1 Satisfaction of Functional Requirements

The TOE Summary Specification Rationale shows, that the set of *TOE Security Functions* (as described in section 7.1) is working together to satisfy the *TOE Security Functional Requirements* (section 6.1). The following table describes the references between the *TOE Security Functional Requirements* (SFR) and the *TOE Security Functions* (<SF1> through <SF4>):

SFR	<SF1>	<SF2>	<SF3>	<SF4>
FCS_CKM.1				X
FCS_CKM.4			X	
FCS_COP.1 (Device encryption 128)		X		
FCS_COP.1 (Device encryption 256)		X		
FCS_COP.1 (Key encryption)	X		X	
FCS_COP.1 (RSA operation)	X			
FCS_COP.1 (Key extraction)	X			
FCS_RND.1				X
FDP_ACC.1		X		
FDP_ACF.1		X		
FIA_SOS.1	X			
FIA_UID.2	X			
FIA_UAU.2	X			
FMT_SMF.1	X		X	
FMT_SMR.1	X			
FMT_MOF.1			X	
FMT_MTD.1	X			
FPT_ITC.1			X	

FCS_CKM.1
The generation of the device encryption key for protection of the user data and the

generation of other keys used for the protection of the TSF data is provided by an internal key generator based on a proprietary digital random number generator (security function <SF4>).

FCS_CKM.4
The overwriting of the (encrypted) system key and the (encrypted) device encryption key is done during uninstallation of the TOE (security function <SF3>).

FCS_COP.1(Device encryption 128) and (Device encryption 256)
The cryptographic operations for encrypting and decrypting user data are included according to AES standard into the driver components and transparently encrypt and decrypt user data on the hard disk as a part of <SF2>.

FCS_COP.1(Key encryption)
The cryptographic operation for securing TSF data is included into the administration program and encrypts the TSF data in the system kernel on the hard disk; this is part of <SF1> and <SF3>.

FCS_COP.1(RSA operation)
If the user requests identification and authentication (<SF1>) with the help of a crypto token, this cryptographic operation is used to analyse the certificate on the token.

FCS_COP.1(Key extraction)
For the storage of keys in encrypted archives this cryptographic operation is used. The keys have to be provided for device encryption/decryption at identification and authentication (<SF1>).

FCS_RND.1
The random number generator provides random numbers, which are then used as input for the internal key generator (<SF4>).

FDP_ACC.1 and FDP_ACF.1
The access control function according to the TOE's security policy is the major consequence of the device encryption function (<SF2>).

FIA_SOS.1
Any new password as entered during the operation of the TOE (<SF1>) is checked against the defined password rules.

FIA_UID.2 and FIA_UAU.2
During the identification and authentication process (<SF1>), the user has to enter a valid user name and the corresponding password or respectively use the secret of his eToken. Otherwise the device encryption key cannot be gained and the operating system can not be booted. The identification and authentication must be done prior to any other system operation.

FMT_SMF.1
The TOE provides functions for the management of the security attributes and the TSF data by retrieving these data from the remote administration as part of <SF3>. Additionally, passwords can be changed during operation of the TOE (<SF1>).

FMT_SMR.1
As stated within the definition of subject <S.USER>, there is only one role maintained at the local user interface of the TOE, which is the *authorised user*. Each user logging in successfully at POA (<SF1>) is assigned to this role. Each user successfully authenticated by certificate at the remote administrative interface is assigned the role *administrative server*.

FMT_MOF.1
Disabling the TOE by uninstalling it is restricted to the *authorised user* (<SF3>).

FMT_MTD.1
The password of a user can only be changed by the user himself (<SF1>).

FPT_ITC.1
The TOE's function for securing remote administration data <SF3> provides the mechanisms for achieving the confidentiality of TSF data exported from and imported to the TOE.

9.4.2 Mutual Support of Security Functions

The security functions <SF1>, <SF2>, <SF3> and <SF4> mutually support each other.

They support the security role *authorised user*. Only authorised users can pass *Power On Authentication (POA) <SF1>*, which is the only instance to provide user certificates and then device encryption keys. These encryption keys are required to operate *Protection of Data on Protected Devices <SF2>* correctly and give the user access to the user data on each device controlled by the TOE.

The security function *Secure Server-Based Administration <SF3>* supports all other security functions. Administrative operations can only be performed by the administration process, as described in <SF3> or (for some functions) by the role *authorised user*, which is guaranteed by *Power On Authentication (POA) <SF1>*.

The security function *Random Number Generation and Key Generation <SF4>* supports the security function *Protection of Data on Protected Devices <SF2>* by providing keys generated out of random numbers.

As shown in the previous section, there is a complete and sufficient mapping between the Security Functions and the Security Functional Requirements. Therefore no additional functionality is required to meet the Security Functional Requirements of the TOE.

9.4.3 TOE Assurance Requirements

The *TOE Security Functional Requirements* (section 6.1) cover all aspects to ensure that the security functions provided by the TOE are actually able to response to the security problems defined in form of *TOE Security Objectives* (section 5.1). The assurance requirements cover those defined for the Evaluation Assurance Level 4. The documentation provided by the sponsor as listed in the table in section 6.2 describes, that the assurance requirements are properly fulfilled.

The TOE itself does not provide any measure or mechanism to satisfy the assurance requirements.

9.5 PP Claims Rationale

This Security Target does not make any claim that the TOE is conformant with the requirements of a *Protection Profile*. As a consequence, the chapter *PP Claims Rationale* is empty.

10 Terms and Definitions

10.1 Abbreviations

AES	Advanced Encryption Standard
BIOS	Basic Input Output System
CC	Common Criteria
DES	Data Encryption Standard
DLL	Dynamically Loadable Library
EFS	Encrypting File System (Microsoft Windows XP and later enhanced NTFS)
FAT	File Access Table (Microsoft DOS/Windows File System)
LAN	Local Area Network
MO	Magneto-Optical device
NTFS	New Technology File System (Microsoft Windows 2000/XP File System)
OS	Operating System
POA	Power-On Authentication
PP	Protection Profile
Rijndael	Symmetric encryption algorithm (used in AES)
SFR	Security Functional Requirement
SGN	SafeGuard Enterprise
SOF	Strength of Function
SSL	Secure Socket Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSF	TOE Security Function
WAN	Wide Area Network

10.2 Definitions

Term	Definition
Operational State	The PC is in operational state, after an authorised (<S.AUTH>) individual has performed login, until the moment where the operating system has been shut down and the PC has been physically switched off. In particular, hibernated PCs are considered to be not in operational state.
Protected Mode	**Protected mode**, also called **protected virtual address mode**, is an operational mode of x86-compatible central processing units (CPU). It was first added to the x86 architecture in 1982, with the release of Intel's 80286 (286) processor and later extended with the release of the 80386 (386) in 1985. Protected mode allows system software to utilize features such as virtual memory, paging, *safe* multi-tasking, and other features designed to increase an operating system's control over application software. When a processor that supports x86 protected mode is powered on, it begins executing instructions in real mode, in order to maintain backwards compatibility with earlier x86 processors. Protected mode may only be entered after the system software sets up several descriptor tables and enables the Protection Enable (PE) bit in the Control Register 0 (CR0).
Real Mode	**Real mode**, also called **real address mode** or **compatibility mode**, is an operating mode of 80286 and later x86-compatible CPUs. All x86 CPUs in the 80286 series and later start in real mode at power-on; 80186 CPUs and earlier had only one operational mode, which is equivalent to real mode in later chips.
Token	The term *token* here often designates both USB tokens and smart cards, basically meaning “authentication device”.
Crypto Token	This designates USB token or smart card that can do cryptographic operations such as AES or RSA on-chip. These tokens usually contain certificates and matching private keys. We distinguish these from other tokens that do not offer cryptographic mechanisms but can e.g. store user ID and password, protected by the token’s PIN.

11 References

[CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; Version 2.3, August 2005

[CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Requirements; Version 2.3, August 2005

[CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Requirements; Version 2.3, August 2005

[AIS20] Application Notes and Interpretation of the Scheme (AIS) 20, Functionality classes and evaluation methodology for deterministic random number generators, Bundesamt für Sicherheit in der Informationstechnik, Version 1, 02 December 1999

[AIS31] Application Notes and Interpretation of the Scheme (AIS) 31, Functionality classes and evaluation methodology for deterministic random number generators, Bundesamt für Sicherheit in der Informationstechnik, Version 1, 25 September 2001

[INSTALL] SafeGuard® Enterprise Version 5.60
Installation Guide, Sophos Group, April 2011 (English Version, PDF file)

[CERTCOM] SafeGuard® Enterprise Version 5.60
Manual for Certification Compliant Operation, Sophos Group, September 2011, PDF file

12 Annex A: SFR Family FCS_RND

This Security Target uses the component FCS_RND.1 as specified in the German certification scheme AIS31, issued by BSI in September 2001.

The definition of the FCS_RND family is cited below.

Originally the definition has been intended for use with “true physical random number generators”, but it can also be used for deterministic random number generators (as done in this document).

12.1 FCS_RND generation of random numbers

Family behaviour

This family defines quality metrics for generating random numbers intended for cryptographic purposes.

Component levelling

FCS_RND.1 The generation of random numbers using TSFs requires the random numbers to meet the defined quality metrics.

Management: FCS_RND.1
No management functions are provided for.

Logging: FCS_RND.1
There are no events identified that should be auditable if FCS_RND generation of random numbers data generation is included in the PP/ST.

FSC_RND.1 Quality metrics for random numbers
Is hierarchical to: no other components.

FCS_RND.1.1 The TSFs shall provide a mechanism for generating random numbers that meet [assignment: *a defined quality metric*].

FCS_RND.1.2 The TSFs shall be able to enforce the use of TSF-generated random numbers for [assignment: list of TSF functions].

Dependencies: FPT_TST.1 TSF testing.