# SafeBoot N.V.

# SafeBoot® Device Encryption™ for PC Version 5.0

# Common Criteria

# Security Target

# Table of Contents

# Table of Figures

## References


| | |
|---|---|
| CC | Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005 (aligned with ISO 15408). |
| FIPS-PUB 180 | Federal Information Processing Standard Publication (FIPS-PUB) 180-1, Secure Hash Standard, 17 April 1995 |
| FIPS-PUB 186 | Federal Information Processing Standard Publication (FIPS-PUB) 186-2, Digital Signature Standard (DSS), 5 October 2001 |
| FIPS-PUB 197 | Federal Information Processing Standard Publication (FIPS-PUB) 197, Advanced Encryption Standard (AES), 26 November 2001 |
| FIPS-PUB 140 | Federal Information Processing Standard Publication (FIPS-PUB) 140-2, Including Change Notices, Security Requirements for Cryptographic Modules, 3 December 2002 |
| SafeBoot Administrators Guide | Device Encryption 5 PC Administrators Guide<br>Version: 2006/09 |
| RFC 2898 | PKCS #5: Password-Based Cryptography Specification Version 2.0, September 2000 |

## Glossary

| | |
|---|---|
| Administration server | A software installation consisting of SBAdmin, SBServer and the SafeBoot Object Directory, all at version 5.0 |
| AES | Advanced Encryption Standard |
| Authorised Administrator | Any entity that is able to establish a secure management session with the TOE |
| Authorised User | Any entity that has logged on to the TOE through the logon GUI |
| CC | Common Criteria |
| CSP | Critical Security Parameters |
| DLL | Dynamic Link Library |
| DSA | Digital Signature Algorithm |
| DSS | Digital Signature Standard |
| EAL | Evaluation Assurance Level |
| FIPS | Federal Information Processing Standard |
| GUI | Graphical User Interface |
| IPC | Inter-process communication |
| IT | Information Technology |
| Machine | The TOE PC |
| MBR | Master Boot Record |
| OS | Operating System |
| PKCS-5 | Public Key Cryptography Standard 5 (Password-Based Cryptography Specification) |
| SAR | Security Assurance Requirement |
| SFP | Security Function Policy |
| SFR | Security Functional Requirement |
| SHA-1 | Secure Hash Algorithm |
| SOF | Strength of Function |
| ST | Security Target |
| TCP/IP | Transmission Control Protocol/Internet Protocol |
| TOE | Target of Evaluation |
| TOE data | The encrypted contents of the TOE hard disk |
| TSC | TSF Scope of Control |

| | |
|---|---|
| TSF | TOE Security Functions |
| TSP | TOE Security Policy |

# 1 Security Target Introduction

## 1.1 Security Target Identification

Security Target Title: SafeBoot N.V. SafeBoot® Device Encryption™ for PC Version 5.0 Common Criteria Security Target.

Security Target Version: 1.0.

TOE Identification: SafeBoot® Device Encryption™ for PC Version 5.0.

Evaluation Assurance Level (EAL): EAL4.

Common Criteria Identification: Common Criteria for Information Technology Security Evaluation, Version v2.3 August 2005, ISO/IEC 15408:2005 and ISO/IEC 18405:2005.

Keywords: disk encryption, access control, security target, EAL4, SafeBoot.

## 1.2 Security Target Overview

SafeBoot Device Encryption for PC is a Personal Computer (PC) security system that prevents the data stored on a PC's hard disk from being read or used by an unauthorized person. It combines single sign-on user access control with transparent full hard disk encryption to offer effective security for PCs running the Microsoft Windows™ operating system.

Management, deployment and user recovery are handled by a centralised administration server and communication between the SafeBoot Device Encryption Client and this administrative server is via TCP/IP using a cryptographically secure proprietary protocol.

## 1.3 Common Criteria Conformance Claim

The identified TOE conforms to the following specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.3, August 2005, ISO/IEC 15408-2.
    - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.3, August 2005, ISO/IEC 15408-3.
    - Part 3 Conformant
    - Evaluation Assurance Level 4 (EAL4)

# 2 TOE Description

SafeBoot Device Encryption for PC is a Personal Computer (PC) security system that prevents the data stored on a PC's hard disk from being read or used by an unauthorized person. In simple terms, the SafeBoot Device Encryption Client takes control of a user's hard disk away from the operating system. The SafeBoot Device Encryption Client encrypts data written to the disk, and decrypts data read from the disk. If the hard disk drive is read directly, one would find only encrypted data, even in the Windows swap file and temporary file areas.

## 2.1 Scope and boundaries of the TOE

The physical boundary of the TOE is the case of the PC on which it is installed.

The logical boundary of the TOE is the application software that corresponds to version 5.0 of the SafeBoot Device Encryption Client. See Figure 1 below.

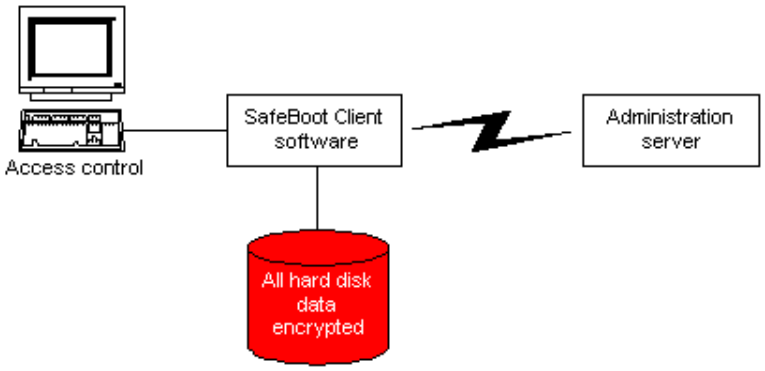


**Figure 1 TOE logical boundary**

At the TOE boundary are its interfaces. There is a man-machine access control interface to allow a user to submit logon credentials for authentication. There is a disk drive interface to allow the contents of the disk drives to be secured through encryption, and there is a secure management interface to allow secure communication with other SafeBoot Device Encryption for PC products within the IT environment.

The IT environment of the TOE includes a PC running either the Microsoft Windows 2000 Professional or Windows XP Professional operating system. Also within the IT environment are the SafeBoot Administrator (SBAdmin), SafeBoot Server (SBServer) and the SafeBoot Object Directory. These entities run on a remote PC or PCs connected to the SafeBoot Device Encryption PC over a network running the TCP/IP protocol. See Figure 2. Because these entities (or equivalent) are crucial to the correct operation of the TOE, assumptions relevant to the IT environment and resulting SFRs are provided for the operating system and Administration server. For the purposes of this evaluation, although other options are feasible, the Administration server consists of SBAdmin, SBServer and the SafeBoot Object Directory, all at version 5.0.

Within the scope of the EAL4 evaluation of the SafeBoot Device Encryption Client, the Administration server is exercised as part of evaluation testing in order to provide assurance as to its effectiveness and to provide assurance that it supports the security of the TOE.

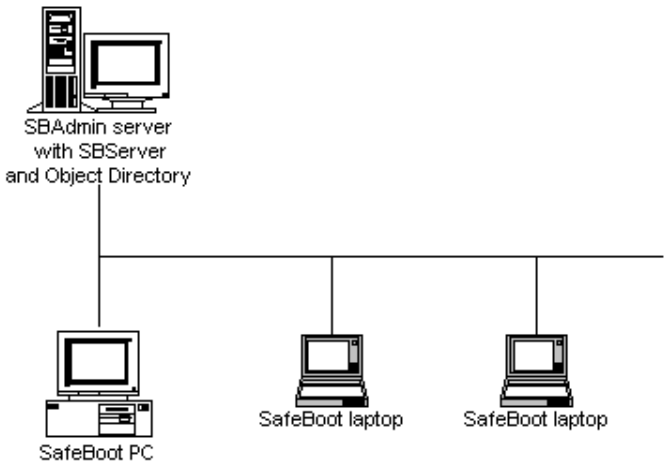


**Figure 2 TOE IT environment**

## 2.2 SafeBoot Device Encryption for PC Family Functional Overview

SafeBoot Device Encryption for PC replaces the boot sector of the hard disk to provide effective access control and optionally encrypts part or all of the hard disk drive. The TOE is a software module running on a standard PC. The processor of the PC executes all software. All software components of the TOE are stored on the hard disk, and, while executing, are stored in the RAM.

SafeBoot Device Encryption for PC supports centralized management of SafeBoot Device Encryption for PC protected machines. SafeBoot Device Encryption for PC components include the SafeBoot Administrator, SafeBoot Server, SafeBoot Object Database, SafeBoot Device Encryption Client, SafeBoot File Encryptor and SafeBoot Connector Manager. Every time a SafeBoot Device Encryption for PC protected machine boots, and optionally every time the user initiates a dial-up connection or after a set period of time, the SafeBoot Device Encryption Client tries to contact its "Object DataBase". This is a central store of configuration information for both machines and users, and is managed by SafeBoot Administrators. The Object DataBase could be on the user's local hard disk (if the user is working completely stand-alone), or could be in some remote location and accessed over Transmission Control Protocol/Internet Protocol (TCP/IP) via a secure SafeBoot Server (in the case of a centrally managed enterprise).

The SafeBoot Device Encryption for PC protected machine queries the Object Database for any updates to its configuration, and if needed downloads and applies them. Typical updates could be a new user assigned to the machine by an administrator, a change in password policy, or an upgrade to the SafeBoot Device Encryption Client operating system or a new file specified by the administrator. At the same time the SafeBoot Device Encryption Client uploads details like the latest audit information, any user password changes, and security breaches to the Object DataBase. In this way, transparent synchronization of the enterprise becomes possible.

SafeBoot Device Encryption for PC has the option of being configured in different ways. At installation, the SafeBoot Administrator can specify how the hard disk can be encrypted by choosing one of three encryption modes: full, partial, or none. Full encryption mode encrypts an entire partition. Partial encryption mode encrypts only a portion of a partition or hard disk. None encryption mode leaves the partition in plaintext with no encryption. Full encryption is the only valid mode that can be used if SafeBoot Device Encryption for PC is to operate in a Common Criteria compliant mode (CC mode) and so comply with the requirements of this Security Target.

CC mode is defined as:

- Password restrictions
  - Minimum password length of five characters
  - Invalidate user's password after ten successive unsuccessful logon attempts
- Full encryption of hard disk
- Users forced to logon
- SafeBoot Device Encryption client screen saver

In order to provide a TOE that is CC mode compliant, the TOE must be configured appropriately. This requires the Administration server to be used in a "CC mode" to deliver a CC-compliant TOE. Details of the method of use to achieve this are provided in the SafeBoot Administrators Guide.

## 2.3 SafeBoot Device Encryption Client

The SafeBoot Device Encryption Client consists of a boot Operating System (OS) (the SafeBoot Device Encryption Client OS), a Basic Input Output System (BIOS) hook, Windows drivers, a system tray application and a set of Windows Dynamic Link Libraries (DLLs). These components comprise the validated TOE. SafeBoot Device Encryption for PC installs a mini-operating system on the user's hard drive, this is what the user sees when they switch on the TOE. SafeBoot Device Encryption for PC looks and feels like Microsoft Windows, with mouse and keyboard support, moveable windows etc. The SafeBoot Device Encryption Client OS is completely self-contained and does not need to access any other files or programs on the hard disk, and is responsible for allowing the user to authenticate.

Once the user has entered the correct authentication information, the SafeBoot Device Encryption Client operating system starts a driver in memory and boots the protected machine's original operating system. From this point on the machine will look and behave as if SafeBoot Device Encryption for PC was not installed.

## 2.4 TOE Interfaces

The TOE includes a computer running an operating system (OS) and interfacing with the computer keyboard, mouse, screen, LAN ports, floppy drive, CD-ROM drive, speaker, disk drive, microphone inputs, serial ports, parallel ports, and power plug.

SafeBoot Device Encryption for PC provides a logical interface via an Application Programming Interface (API) and a Graphical User Interface (GUI). This logical interface exposes services (described in section 2.6) that the User, the operating system and SafeBoot Device Encryption Client applications may utilize directly.

The logical interfaces provided by the SafeBoot Device Encryption Client are: data input, data output, control input, and status output as follows:

- Data Input – Input to all driver functions
- Data Output – Output from all driver functions
- Control Input – Input from TCP/IP interface, IPC interface, GUI
- Status Output – Return codes from driver functions, Show Status GUI option

The Data Input and Data Output interfaces are the interfaces through which data is encrypted with the chosen algorithm (more information found in section 2.8) prior to being written to a disk and encrypted data is decrypted when read from a disk.

The Control Input interface is the means by which the client is configured. This is the secure management interface provided by the TOE and driven by the Administration server. All configuration information is applied via synchronization operations with the associated Object Database. Synchronization can be initiated by several means, including: TCP/IP connections to/from the management software, IPC (inter-process communications) functions and GUI options on the system tray application.

The Status Output interface consists of text information displayed in a dialog box when the "Show Status" option is selected on the system tray application menu.

## 2.5 Operational Environment

The TOE runs on a standard IBM-compatible personal computer running a variant of the Windows operating system (the TOE is being evaluated on the Windows 2000 and XP platforms).

The TOE runs in its own operating system threads. This provides it with protection from all other processes, preventing access to all keys, intermediate key generation values and other CSPs.

The task scheduler and architecture of the operating system maintain the integrity of the TOE.

The TOE is protected using password-based access control. The driver files are themselves encrypted on the hard drive of the PC. So, any attacker would either need to possess the appropriate password, or would need to be able to decrypt the hard drive to gain access to the executable code of the TOE. The source code is not included as part of the TOE, and the keys and CSPs are not stored in a plaintext form on the TOE.

There is no upper limit to the number of Users of the TOE, although only one operator can have access to the PC that contains the TOE at a time.

## 2.6 Roles and Services

The SafeBoot Device Encryption Client implements two roles: an Administrator role and a User role. The TOE performs identity-based authentication for Users and role-based authentication for Administrators.

The following table, Figure 3, summarizes the services available to each role.

| Role | Purpose | Services |
|---|---|---|
| Administrator | TOE configuration | - Connect to TOE using the Administration server, via an encrypted session to transmit control data |
| User | Usage of TOE functionality | - Utilize hard disk encryption services<br>- Initiate synchronization with management software<br>- View status |

**Figure 3 Roles**

The User role is assumed when a SafeBoot Device Encryption for PC protected machine is booted and proper username and password is entered into the login prompt displayed by the boot SafeBoot Device Encryption Client OS. Once authenticated, user specific information and key material are loaded from the SBFS (SafeBoot File System) and the original operating system (with SafeBoot Device Encryption for PC drivers installed) is launched. The necessary key material and machine state information is loaded into the drivers and the transparent encryption/decryption of disk-based information begins. A system tray application, which may be configured to start automatically, may be used to view the status of the TOE or to initiate a synchronization operation.

The Administrator role may be assumed by establishing an authenticated encrypted session with the SafeBoot Device Encryption Client for purposes of configuring the TOE. The user interface is provided by the Administration server. All communications between the management software and the client are encrypted using AES (with a session key generated using Diffie-Hellman key agreement). DSA is also used during the Diffie-Hellman key agreement to authenticate the server to prevent server spoofing.

Figure 4 summarizes the authentication mechanism for each of these roles, and Figure 5 describes the strength of these mechanisms.

| Role | Type of Authentication | Authentication Data |
|---|---|---|
| User | Identity-based | Password |
| Administrator | Role-based | DSS authenticated challenge-response mechanism |

**Figure 4 Roles and Required Identification and Authentication**

| Authentication Mechanism | Strength of Mechanism |
|---|---|
| Password | It is possible to configure the minimum password length and the type of characters that can be used in a password. It is also possible to configure the client to lock up after a specified number of unsuccessful password entry attempts. SafeBoot recommends a minimum password length of 5 characters, giving a random chance of success of 1 in 916,132,832. The software is configured to lock up after 10 unsuccessful attempts; this gives a chance of successfully guessing the password at 1 in 91,613,283. |
| DSS authenticated challenge-response mechanism | Given that the key size is 1024 bits, that amounts to $2^{1024}$ (approximately $1.8 \times 10^{308}$ possible keys). |

**Figure 5 Strength of Authentication Mechanisms**

## 2.7 Access to Services

The following table, Figure 6, lists the authorized services linked to each of the Roles offered by the TOE. For an explanation of the various services listed, the reader should consult the SafeBoot Administrators Guide.

| Role | Authorized Services |
|---|---|
| User (via TOE UI) | Synchronization |
| | Encryption/Decryption |
| | Show Status Functions |
| | Self-test Functions |
| | Change User Password |
| Administrator (via Administration server UI driving the TOE management interface) | Synchronization |
| | Show Status Functions |
| | User/machine recovery requests |
| | Configuration |
| | File Updates |
| | Manage SafeBoot Device Encryption for PC (Cryptographic & Key Management Functions) |
| | Software Updates |
| | User/machine recovery |
| | Set User Attributes (passwords, access rights, etc.) |
| | Change User Attributes |
| | Create User Groups |
| | Modify User Groups |
| | Delete User Groups |
| | Create Users |
| | Modify Users |
| | Delete Users |

**Figure 6 Services Authorized for Roles**

## 2.8 Cryptographic Key Management

The TOE uses a variety of keys, including: hard disk encryption key, user encryption keys, session keys, recovery keys, database key (when used with a local Object Database only), integrity check keys and server public key. The following table, Figure 7, lists all keys. Currently, AES is the only approved encryption algorithm in the SafeBoot Device Encryption Client product and all encryption keys are AES keys. The server public key is a DSA key.

| Key type | Purpose |
|---|---|
| Hard disk encryption key | To encrypt hard disk contents; to authenticate client to the Object Database; to encrypt database key (when local Object Database is used) |
| User encryption keys | To encrypt secure user attributes |
| Server public key | To authenticate the Administrator communications |
| Machine recovery key | Encryption key used to recover the hard disk |
| User recovery keys | To recover user encryption keys |
| Database key | Used only with local Object Database to protect certain attributes |
| Integrity check key | Used to perform TOE integrity check |
| Session keys | To encrypt traffic between client and remote server |
| Diffie-Hellman Keys | Used to establish session keys |
| SHA-1 Known Answer Test (KAT) parameters<br>DSA Test KAT parameters<br>AES KAT data | A fixed set of parameters used to validate the SHA-1 functionality during the SHA-1 known answer test performed at power-up<br>A fixed set of parameters used to validate the DSA functionality during the DSA known answer test performed at power-up<br>Test data is taken from NIST Special Publication 800-38A 2001 Edition "Recommendations for Block Cipher Modes of Operation" This uses fixed parameters to generate cipher text and plain text which is then verified against expected values |

| Key type | Purpose |
|---|---|
| PRNG seed values and seed keys | These are used to prevent the output from the PRNG from being predictable to an attacker. Knowledge of the seed values and seed keys is required to predict key values. |
| PRNG KAT data | Used to test the pseudo-random number generator to verify that it is operating correctly |

**Figure 7 Keys used by SafeBoot Device Encryption Client**

### 2.8.1 Key generation

The SafeBoot Device Encryption Client generates symmetric key material (and the Diffie-Hellman public/private key pair used in session key establishment,) using a FIPS 186-2 Appendix 3.3 compliant pseudo-random number generator. The only symmetric keys generated in this way are the Hard Disk Encryption Key, the Machine Recovery Key and the Session Keys.

### 2.8.2 Key entry and output

All key material, excluding recovery key information and the Diffie-Hellman public key used in session key establishment, is entered and output from the TOE in encrypted form. Recovery key information can be entered manually in plaintext form, electronically in plaintext. When entered manually, correct key entry is verified using a checksum.

### 2.8.3 Key storage

Key material is stored in the SafeBoot File System (SBFS). All key material is encrypted using AES prior to storage. All sectors of the SBFS feature a checksum to guard against modification.

### 2.8.4 Protection of key material

The SafeBoot Device Encryption Client securely manages key material for the lifetime of the key. All key material is encrypted with AES prior to storage in the SBFS and prior to export.

### 2.8.5 Zeroization of key material

All key material mentioned in Figure 7 above (the complete list of unprotected critical security parameters - CSPs), associated with a machine is zeroized when the SafeBoot Device Encryption Client is uninstalled. All user encryption key material associated with users is zeroized when the user is deleted.

## 2.9 Cryptographic Algorithms

The SafeBoot Device Encryption Client supports the following algorithms:

- AES,
- DSA
- SHA-1.
- Diffie-Hellman

## 2.10 Self-Tests

The SafeBoot Device Encryption Client implements both power-up and conditional self tests. The following two sections outline the tests that are performed.

## 2.11 Power-up self-tests

The following table, Figure 8, lists the power-up self-tests performed by the TOE:

| |
|---|
| *SHA-1 known answer test* |
| *DSA known answer test* |
| *AES known answer test* |
| *Critical Functions (Configuration file signature verification test)* |

| |
|---|
| *Software/Firmware integrity test (Signature verification)* |
| *Pseudo-Random Number Generator Known Answer Test* |

**Figure 8 Power-up self-tests**

Each of these tests is executed when the computer is turned on and the TOE first executes. If any of these tests fail, the TOE will not load. The TOE must be reset to re-execute these tests.

### 2.11.1 Conditional self-tests

There are three conditional tests that are run by the TOE. A continuous random number generator test is run every time the TOE requests a random number. Failure of this test may result in keys not being generated and an appropriate error message will be given. A test is also done when a software update occurs. All files are digitally signed and this signature is checked prior to any update of the software. There is also a manual key entry test that verifies correct entry of the user recovery keys and machine recovery key. More information on this test can be found in chapter 18 of the Administrator's Guide.

# 3 TOE Security Environment

## 3.1 Assumptions

This section describes the assumptions that have been made about the environment in which the TOE is used, including assumptions about personnel and the physical environment of the TOE. The TOE operates in a secure manner and provides its countermeasures as long as it is utilised in a manner that adheres to the intended environment, and method of delivery, installation and administration.

### 3.1.1 Personnel Assumptions

This section describes the assumptions about how the staff that are authorised to use the TOE behave.

A.MANAGEMENT
One or more proficient persons are assigned to administer the TOE and the security its data.

A.NO_MALEVOLENCE
The system administrators are not careless, malicious or intentionally negligent, and can be expected to follow the administrative guidance given to them in the TOE administration documentation.

A.PROFICIENT_USERS
Authorised TOE users and administrators follow the guidance provided for the secure operation of the TOE. There is no formal user guidance, it is the responsibility of the administrator to ensure that the users that he is responsible for are given appropriate guidance.

A.AUTHENTICATION_DATA_PRIVATE
Authentication data is kept private by authorised users of the TOE.

### 3.1.2 Physical Assumptions

This section describes the assumptions made about the physical environment in which the TOE operates.

A.TIME_SOURCE
The TOE's IT environment provides a reliable time source to enable the TOE to timestamp audit records.

A.ADMINISTRATION_SERVER
The TOE's IT environment provides an administration server to facilitate effective management of the security functions of the TOE. This server provides client PCs with a central point for storage of installation files, recovery files, update profiles, and software updates.

A.SECURE_BACKUP
User's data backups are separately encrypted or physically protected to ensure data security is not compromised through theft of or unauthorised access to backup information.

A.AVAILABLE_BACKUP
Regular and complete backups are taken to enable recovery of user data in the event of loss or damage to data as a result of the actions of a threat agent.

A.DOMAIN_SEPARATION
The operating system is able to provide separate threads of execution to protect the TOE from interference from other software running on the TOE PC.

A.TRUSTED_SOFTWARE
The software environment runs only trusted software that has been approved by the network manager. This also presumes appropriate protections against malicious installation of non-approved software such as viruses and Trojan horses by the appropriate deployment of firewalls, bastion hosts, and anti-virus software as appropriate.

### 3.1.3 System Assumptions

This section describes the assumptions made about the whole of the system of which the TOE forms a component. The assumptions are made in relation to the TOE.

A.NON_TECHNICAL_IDENTITY_VERIFICATION
There is a database of authorised TOE-users along with user-specific authentication data for the purpose of enabling administrative personnel to verify the identity of a user over a voice-only telephone line before providing them with support.

## 3.2 Threats

This section describes the threats to the assets of the TOE against which specific protection within the TOE or its environment is required.

This section describes the threat profilethat the TOE addresses. This profile should be considered in the context of a global system security policy. The TOE is a PC access control and hard disk encryption product and the threats it addresses are selected in order to fulfil these objectives.

T.ACCESS
An unauthorised user of the TOE may access information without having permission from the person who owns, or is responsible for, the information. This threat is applicable if the TOE is stolen or otherwise falls into the hands of an attacker who then attempts to gain unauthorised access to the assets protected by the TOE.

T.ALTERNATE_BOOT_PROCESS
An unauthorised user with physical access to the system may use a boot floppy or similar device to subvert the system's normal boot process in order to access information assets contained on the system.

T.CONFIG _MODIFICATION
Configuration data or other sensitive data (such as registry settings) may be modified by unauthorised users.

T. CORRUPT_AUDIT
Unauthorised users may modify audit data by gaining unauthorised access to the audit trail.

T.EASE_OF_USE_ADMIN
The administrator may unintentionally select insecure configuration parameters or insecure default configuration parameters for the user.

T.EASE_OF_USE_USER
The user may unintentionally select insecure configuration parameters, reducing the security of the TOE.

T.EAVESDROP_TRANSIT
An unauthorised user may listen in on communications (electronic or otherwise) between the TOE and an administration server, and so gain unauthorised access to informaton.

T.OBJECT_REUSE
Using expired authentication data, users may gain unauthorised access to information.

T.PASSWORD_LOSS
The user may forget their password, making data unavailable.

T.RECORD_ACTIONS
An unauthorised user may perform unauthorised actions that go undetected.

T.RECOVERY_PROCEDURE_INTERCEPT
An unauthorised user may eavesdrop on telephone communications between user of the TOE and the help desk when a user is performing the recovery procedure, and so gain unauthorised access to information.

T.RECOVERY_MASQUERADE
An unauthorised user with physical access to the TOE may try and perform the recovery procedure in order to gain access to the information securely stored on the TOE.

T.REMOVE_DISK
An unauthorised user with physical access to the system may remove a hard drive from the system in order to circumvent the authentication mechanisms of the TOE and gain access to information contained on the hard drive.

T.SPOOF
A hostile entity may impersonate the TOE in order to gain unauthorised access to authentication data, such as by presenting a look-alike logon screen and asking for the user's password.

T.SYSTEM_ACCESS
An unauthorised user may gain unauthorised access to the system and act as an administrator or other authorised user.

T.UNAUTHORISED_MODIFICATION
An unauthorised user may modify the TOE software (executable code), and so gain unauthorised access to system and user resources.

## 3.3 Organisational Security Policies

This section describes the complete set of organisational security policy statements or rules with which the TOE must comply.

P.AUTHORISED_USERS
Only authorised users may use the system.

P.CRYPTOGRAPHIC_KEYS
Cryptographic keys will be generated, accessed, protected, and destroyed in accordance with requirements defined by FIPS 140-2 Level 1.

P.CRYPTOGRAPHIC_OPERATIONS
All cryptographic operations performed by the system will be compliant with the requirements of FIPS 140-2 level 1 and FIPS 197 (AES).

P.EAVESDROP_TRANSIT
System data must be protected in transmission between the protected system and the administration server.

P.FAULT_TOLERANCE
Access control functions must be able to continue to operate if systems lose communications with central administration servers.

P.USER_ACCOUNTABILITY
Users of the system shall be held accountable for their security relevant actions within the system.

# 4 Security Objectives

Security objectives address all of the security environment aspects identified. They reflect the intended method of use of the TOE and are suitable to counter all identified threats and cover all identified organisational security policies and assumptions.

## 4.1 Objectives for the TOE

O.AUTHORISATION
The TSF must ensure that only authorised users gain access to the TOE and its resources by uniquely identifying all users and authenticating their claimed identity before granting access to the TOE and its resources.

O.ACCESS_CONTROL
The TSF must control access to the TOE based on user identity. The TSF must provide the ability to limit each user's access.

O.ENCRYPTED_MEDIA
The TSF must provide encryption to protect designated information assets from unauthorised users that have gained physical access to the TOE's storage media.

O.EFFECTIVE_ADMINISTRATION
The TOE must allow administrators to effectively manage the TOE and its security functions, and must ensure that only authorised administrators are able to access such functionality.

O.AUDIT
The TSF must record the security relevant actions of users of the TOE and have the ability to associate each action with an identified user where possible. The TSF must present this information in a comprehensible format to authorised users while preventing access to unauthorised users.

O.SECURE_RECOVERY
The TOE should allow the user with assistance from an administrator to regain access to his machine and set a new password after forgetting his password.

O.PROTECT
The TSF must protect its own data and resources. It must protect against external interference or tampering.

O.TRUSTED_PATH
The TSF must provide the capability to allow users to ensure that they are communicating with the TOE during initial authentication and not with another entity impersonating the TOE.

O.DATA_TRANSFER
The TSF must have the capability to protect system data in transmission between any administration server and the TOE

O.CRYPTOGRAPHIC_KEYS
The TSF must ensure that cryptographic keys are generated, accessed, protected, and destroyed in accordance with requirements defined by FIPS 140-2 Level 1.

O.CRYPTOGRAPHIC_OPERATIONS
The TSF must ensure that all cryptographic operations used to protect information and encryption keys are compliant with the standards defined by FIPS 140-2 Level 1 and FIPS 197 (AES).

O.FAULT_TOLERANCE
The TSF must continue to enforce access control policies if communications are lost with the central administration server.

O.EASE_OF_USE_USER
The TSF must prevent the user from configuring the TOE in an insecure fashion. As an aid to this, the user must be allowed to change his password and to clear the audit trail, as required.

O.NO_OBJECT_REUSE
The TSF must prevent users gaining unauthorised access to information using expired authentication data.

## 4.2 Objectives for the Environment

### 4.2.1 Security Objectives for the IT Environment

OE.TIME_SOURCE
The TOE IT environment must provide a reliable time source to enable the TOE to timestamp audit records.

OE.ADMINISTRATION_SERVER
The TOE IT environment must provide an administration server to manage the TOE client software. This server provides client PCs with a central point for storage of ins tallation files, recovery files, update profiles, and software updates.

OE.SECURE_BACKUP
The TOE IT environment must create user data backups that are separately encrypted or physically protected to ensure data security is not compromised through theft of or unauthorised access to backup information.

OE.AVAILABLE_BACKUP
The TOE IT environment must take regular and complete backups are taken to enable recovery of user data in the event of loss or damage to data as a result of the actions of a threat agent.

OE.DOMAIN_SEPARATION
The TOE IT environment must provide separate threads of execution for TOE processes.

OE.TRUSTED_SOFTWARE
The TOE IT environment must run only trusted software that has been approved by the network manager. This also presumes appropriate protections against malicious installation of non-approved software such as viruses and Trojan horses by the appropriate deployment of firewalls, bastion hosts, and anti-virus software as appropriate.

### 4.2.2 Security Objectives for the Non-IT Environment

OE.MANAGED
Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner that maintains IT security objectives. These are competent, trained administrators who are not careless, negligent or hostile.

OE.AUTH
Those responsible for the TOE must ensure that users protect all access credentials, such as passwords or other authentication information in a manner that maintains IT security objectives.

OE.EASE_OF_USE_ADMIN
The administrator should ensure that the TOE is configured securely, that is that the TOE is operating in CC mode.

OE.EASE_OF_USE_USER
The user should ensure that his password is not divulged to an unauthorised party and that the TOE is never left unattended while the user is logged onto it.

OE.NON_TECHNICAL_IDENTITY_VERIFICATION
There is a database of authorised TOE-users along with user-specific authentication data for the purpose of enabling administrative personnel to verify the identity of a user over a voice-only telephone line before pr oviding them with support.

# 5 IT Security Requirements

## 5.1 TOE Security Functional Requirements

This section defines the functional requirements for the TOE in terms of functional components drawn from Part 2 of the Common Criteria. Text contained within square brackets "[]" occurs where selection or assignment operations have been performed within the component.

Note regarding terminology used in assignment operations: For the purposes of this document, an **authorised administrator** is any entity that is able to establish a secure management session with the TOE, and an **(authorised) user** is any entity that is logged on to the TOE through the logon GUI.

An administration server allows an authorised administrator to establish a secure session with the TOE. The existence of this secure session exposes a configuration and control interface to the TOE and the administration server provides a GUI to enable the authorised administrator to make use of this interface. So, when the phrase "authorised administrator" is used with respect to the TOE it refers to any entity that is able to establish a secure management session with the TOE, although in practice this entity is effectively an authenticated user of an Administration server

Because the requirement AVA_SOF.1 is included in the assurance requirements for this TOE (see section 5.2), this document is required to contain a minimum strength of function claim for the security functional requirements.

For this TOE, the minimum strength of function claimed is SOF-medium.

Strength of function only applies to probabilistic or permutation mechanisms that are non-cryptographic. Therefore, the claim here of SOF-medium does not apply to any cryptographic mechanisms with respect to a CC evaluation, and only applies to the password authentication mechanism.

No claims are made about the strength of function of any of the other cryptographic mechanisms that are included in the TOE (disk encryption, key encryption, secure management authentication), and the assessment of algorithmic strength for these mechanisms does not form part of the evaluation.

| Functional Class | Functional Components |
|---|---|
| FAU: Security Audit | FAU_GEN.1 Audit data generation |
| | FAU_GEN.2 User identity association |
| | FAU_STG.1 Protected audit trail storage |
| | FAU_STG.3 Action in case of possible audit data loss |
| FCS: Cryptographic Support | FCS_CKM.1(a) Cryptographic key generation (symmetric) |
| | FCS_CKM.1(b) Cryptographic key generation (asymmetric) |
| | FCS_CKM.4 Cryptographic key destruction |
| | FCS_COP.1(a) Cryptographic operation (data encryption and decryption) |
| | FCS_COP.1(b) Cryptographic operation (key encryption and decryption) |
| | FCS_COP.1(c) Cryptographic operation (authenticated administration) |
| FDP: User data protection | FDP_ACC.2(a) Complete access control (user) |
| | FDP_ACF.1(a) Security attribute based access control (user) |
| FIA: Identification and authentication | FIA_AFL.1 Authentication failure handling (user logon) |
| | FIA_ATD.1 User attribute definition |
| | FIA_UAU.2 User authentication before any action |
| | FIA_UAU.4(a) Single-use authentication mechanisms (secure management) |
| | FIA_UAU.7 Protected authentication feedback |
| | FIA_UID.2 User identification before any action |
| FMT: Security Management | FMT_MSA.1(a) Management of security attributes (secure management) |
| | FMT_MTD.1(a) Management of TSF data (audit) |
| | FMT_MTD.1(b) Management of TSF data (password) |
| | FMT_MTD.2(a) Management of limits on TSF data (authentication |

| Functional Class | Functional Components |
|---|---|
| | failure) |
| | FMT_REV.1(a) Revocation |
| | FMT_SMF.1 Specification of Management Functions |
| | FMT_SAE.1(a) Time-limited authorisation (secure management) |
| | FMT_SMR.1(a) Security roles |
| FPT: Protection of the TSF | FPT_AMT.1 Abstract machine testing |
| | FPT_FLS.1 Failure with preservation of secure state |
| | FPT_RCV.1 Manual recovery |
| | FPT_RVM.1 Non-bypassability of the TSP |
| FRU: Resource utilisation | FRU_FLT.1 Degraded fault tolerance |
| FTA: TOE access | FTA_SSL.2 User-initiated locking |
| | FTA_TSE.1 TOE session establishment |
| FTP: Trusted path/channels | FTP_TRP.1 Trusted path |

**Figure 9 Functional components of the TOE**

### 5.1.1 FAU: Security Audit

FAU_GEN.1 Audit data generation

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [not specified] level of audit; and
c) [All try events, resulting from:
   - Attempts to make changes to passwords
   - Recovery attempts
   - Expiry and timeouts
d) All success events, such as
   - Changes to passwords
   - Logon
   - Recovery
e) All failure events.
   - Password change failures
   - Logon failures
   - Recovery failures]

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [no other relevant information]

**Note**: Auditing is always active in the TOE. Audit entries are not created for the start-up and shutdown of the audit functions as these functions are never started up or shutdown. As long as the TOE is operational, its audit functions are also operational.

FAU_GEN.2 User identity association

FAU_GEN.2.1 The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_STG.1 Protected audit trail storage

FAU_STG.1.1 The TSF shall protect the stored audit records from unauthorised deletion.

FAU_STG.1.2 The TSF shall be able to [prevent] unauthorised modifications to the audit records in the audit trail.

FAU_STG.3 Action in case of possible audit data loss

FAU_STG.3.1 The TSF shall take [the action of overwriting the oldest audit records first] if the audit trail exceeds [3000 items].

### 5.1.2 FCS: Cryptographic Support

The DSA, AES and SHA-1 algorithms are excluded from the scope of the CC TOE. FIPS validation certificates confirm their correct implementation and complement the CC evaluation The TOE has been certified against FIPS 140-2. As part of this process, the cryptographic algorithms used by the TOE were tested and approved. As a result, the TOE contains a number of FIPS-approved algorithms: DSA (FIPS 140 certificates 53 and 112), AES (certificates 21 and 170), and SHA-1 (certificates 71 and 254). These certificates refer to an earlier version of the TOE (version 4.2), but the implementation of these algorithms is identical in the current version, that is, version 5.0.

The TOE generates symmetric keys to use to encrypt the hard disk and asymmetric keys to secure the TOE management protocol and in the process of doing so also generates symmetric key to encrypt these protocol messages once a secure management session has been established.

FCS_CKM.1(a) Cryptographic key generation (symmetric)

FCS_CKM.1.1(a) The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [FIPS 140-2 Compliant Key Generation Algorithm] and specified cryptographic key sizes [256 bits] that meet the following: [FIPS 140-2, Section 4.7.2 Key Generation].

FCS_CKM.1(b) Cryptographic key generation (asymmetric)

FCS_CKM.1.1(b) The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [Diffie-Hellman key exchange algorithm] and specified cryptographic key sizes [1024 bits] that meet the following: [FIPS 140-2, Section 4.7.2 Key Generation].

FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [zeroing] that meets the following: [FIPS 140-2, Section 4.7.6 Key Destruction].

FCS_COP.1(a) Cryptographic operation (data encryption and decryption)
*The TOE uses AES for hard disk encryption*

FCS_COP.1.1(a) The TSF shall perform [data encryption and decryption] in accordance with a specified cryptographic algorithm [AES] and cryptographic key sizes [256 bits] that meet the following: [FIPS 197].

FCS_COP.1(b) Cryptographic operation (key encryption and decryption)

FCS_COP.1.1(b) The TSF shall perform [key encryption and decryption] in accordance with a specified cryptographic algorithm [AES] and cryptographic key sizes [256 bits] that meet the following: [FIPS 197].

FCS_COP.1(c) Cryptographic operation (authenticated administration)

FCS_COP.1.1(c) The TSF shall perform [encrypted and authenticated session based communication with an administration server] in accordance with a specified cryptographic algorithm [AES for encryption and DSA and SHA-1 for authentication] and cryptographic key sizes [256 bits for AES and 1024 bits for DSA] that meet the following: [FIPS 197 for AES and FIPS 186-2 for DSA and SHA-1].

### 5.1.3 FDP: User data protection

FDP_ACC.2(a) Complete access control (user)

FDP_ACC.2.1(a) The TSF shall enforce the [machine access control SFP] on [all users and the TOE data] and all operations among subjects and objects covered by the SFP.

FDP_ACC.2.2(a) The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP.

FDP_ACF.1(a) Security attribute based access control (user)

FDP_ACF.1.1(a) The TSF shall enforce the [machine access control SFP] to objects based on the following: [user identity as defined in the SafeBoot Device Encryption for PC object database and each user's associated password and keys].

FDP_ACF.1.2(a) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [If a user identity has been verified via authentication of the user identity using a supplied password, the derived keys will be used to give the user access to the assets protected by the TSF. Authentication failure will result in the user failing to gain access to the TSF assets].

FDP_ACF.1.3(a) The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [none].

FDP_ACF.1.4(a) The TSF shall explicitly deny access of subjects to objects based on the [none].

### 5.1.4 FIA: Identification and authentication

FIA_AFL.1 Authentication failure handling (user logon)

FIA_AFL.1.1 The TSF shall detect when ["an administrator configurable positive integer within the range 1 to 20"] unsuccessful authentication attempts occur related to [user logon].

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [disable the user account].

FIA_ATD.1 User attribute definition

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users: [

a) User Identitifier
b) Password policy
c) Hard disk encryption key
d) User encryption keys].

FIA_UAU.2 User authentication before any action

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.

FIA_UAU.4(a) Single-use authentication mechanisms (secure management)

FIA_UAU.4.1(a) The TSF shall prevent reuse of authentication data related to [the secure management mechanism and the offline recovery mechanism].

FIA_UAU.7 Protected authentication feedback

FIA_UAU.7 .1 The TSF shall provide only [success-fail feedback in the case of the secure management mechanism and feedback consisting of a '*' for each character typed for all passwords] to the user while the authentication is in progress.

FIA_UID.2 User identification before any action

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.

### 5.1.5 FMT: Security Management

The SFRs in this section refer to the TOE secure management interface. This is the interface that electronically connects the Administration server to the TOE using the secure management protocol.

FMT_MSA.1(a) Management of security attributes (secure management)

FMT_MSA.1.1(a) The TSF shall enforce the [secure management SFP] to restrict the ability to [assign, change_default, query, modify, delete]] the security attributes [all SafeBoot Device Encryption for PC Machine properties and User properties] to [authorised administrators].

FMT_MTD.1(a) Management of TSF data (audit)

FMT_MTD.1.1(a) The TSF shall restrict the ability to [query, clear] the [TSF audit data] to [authorised administrators].

FMT_MTD.1(b) Management of TSF data (password)

FMT_MTD.1.1(b) The TSF shall restrict the ability to [modify] the [a user's password] to [authorised administrators and a user may modify his own password if he successfully supplies his existing password first].

FMT_MTD.2(a) Management of limits on TSF data (authentication failure)

FMT_MTD.2.1(a) The TSF shall restrict the specification of the limits for [successive logon authentication failures] to [authorised administrators].

FMT_MTD.2.2(a) The TSF shall take the following actions, if the TSF data are at, or exceed, the indicated limits: [disable the user account until enabled again by an authorised administrator (as specified by an authorised administrator)].

FMT_REV.1(a) Revocation (secure management)

FMT_REV.1.1(a) The TSF shall restrict the ability to revoke security attributes associated with the [users] within the TSC to [authorised administrators].

FMT_REV.1.2(a) The TSF shall enforce the rules [Revocation can either take place the next time the user logs on, or the user can be revoked immediately, with their machine rebooted and their account disabled or deleted, as specified by the administrator].

FMT_SAE.1(a) Time-limited authorisation (secure management)

FMT_SAE.1.1(a) The TSF shall restrict the capability to specify an expiration time for [user passwords] to [authorised administrators].

FMT_SAE.1.2(a) For each of these security attributes, the TSF shall be able to [give the user a warning that the password is about to expire a specified time before expiry, but also prevent the user from logging on until he has changed the password] after the expiration time for the indicated security attribute has passed.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [User: Changing password of current user. Administrator: Modification of security attributes listed in section 10.]

FMT_SMR.1(a) Security roles (TSF)

FMT_SMR.1.1(a) The TSF shall maintain the roles [Administrator, User].

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

### 5.1.6 FPT: Protection of the TSF

FPT_AMT.1 Abstract machine testing

FPT_AMT.1.1 The TSF shall run a suite of tests [during initial start-up, and in the case of the random number generator test, continuously] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

FPT_FLS.1 Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [unexpected termination of communications with administration server or power failure to TOE PC].

FPT_RCV.1 Manual recovery

FPT_RCV.1.1 After [a user account has been disabled or the user has forgotten their logon password when they try to logon], the TSF shall enter a maintenance mode where the ability to return to a secure state is provided.

FPT_RVM.1 Non-bypassability of the TSP

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

### 5.1.7 FRU: Resource utilisation

FRU_FLT.1 Degraded fault tolerance

FRU_FLT.1.1 The TSF shall ensure the operation of [uninterrupted user access to the TOE if this is allowed within the user configuration] when the following failures occur: [the link to the administration server is lost].

### 5.1.8 FTA: TOE access

FTA_SSL.2 User-initiated locking

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user's own interactive session, by

a) Clearing or overwriting display devices, making the current contents unreadable;
b) Disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 The TSF shall require the following events to occur prior to unlocking the session: [a user to be successfully authenticated via a logon screen by presenting his user identity and password for authentication].

FTA_TSE.1 TOE session establishment

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on [user status. A user whose account is disabled will not be permitted to establish a session].

### 5.1.9 FTP: Trusted path/channels

FTP_TRP.1 Trusted path

FTP_TRP.1.1 The TSF shall provide a communication path between itself and [local] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

FTP_TRP.1.2 The TSF shall permit [local users] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [initial user authentication].

## 5.2 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the Evaluation Assurance Level (EAL) 4 components as specified in Part 3 of the Common Criteria. No operations have been applied to the TOE's assurance components. The table below provides a listing of all Security Assurance Requirements met by the TOE. For a detailed description of these components, please refer to the Common Criteria documentation directly.

| Assurance class | Assurance components |
|---|---|
| Class ACM: Configuration Management | ACM_AUT.1 Partial CM automation<br>ACM_CAP.4 Generation support and acceptance procedures<br>ACM_SCP.2 Problem tracking CM coverage |
| Class ADO: Delivery and Operation | ADO_DEL.2 Detection of modification<br>ADO_IGS.1 Installation, generation, and start-up procedures |
| Class ADV: Development | ADV_FSP.2 Fully defined external interfaces<br>ADV_HLD.2 Security enforcing high-level design<br>ADV_IMP.1 Subset of the implementation of the TSF<br>ADV_LLD.1 Descriptive low-level design<br>ADV_RCR.1 Informal correspondence demonstration<br>ADV_SPM.1 Informal TOE security policy model |
| Class AGD: Guidance documents | AGD_ADM.1 Administrator guidance<br>AGD_USR.1 User guidance |
| Class ALC: Life Cycle Support | ALC_DVS.1 Identification of security measures<br>ALC_LCD.1 Developer defined life-cycle model<br>ALC_TAT.1 Well-defined development tools |
| Class ATE: Tests<br><br><br><br>Class AVA: Vulnerability assessment | ATE_COV.2 Analysis of coverage<br>ATE_DPT.1 Testing: high-level design<br>ATE_FUN.1 Functional testing<br>ATE_IND.2 Independent testing - sample<br>AVA_MSU.2 Validation of analysis<br>AVA_SOF.1 Strength of TOE security function evaluation<br>AVA_VLA.2 Independent vulnerability analysis |

**Figure 10 Assurance Components**

## 5.3 Security Requirements for the IT Environment

This section identifies the IT security requirements that are to be met by the IT environment of the TOE. In this case, the requirements in this part of the Security Target are drawn from Common Criteria Part 2 and as such have been rephrased to clearly indicate that the IT environment, not the TOE, must meet the requirement. Such rephrasing is a special case of refinement and not subject to the assessment requirements associated with modified CC components.

FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The IT environment shall be able to provide reliable time stamps for the use of the TSF. This SFR maps to the assumption A.TIME_SOURCE.

FPT_SEP.1 Domain separation

FPT_SEP.1.1 The IT environment shall maintain a security domain for the execution of the TOE that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The IT environment shall enforce separation between the security domains of subjects in the TSC.

FMT_MSA.2(a) Secure security attributes (secure management)

FMT_MSA.2.1(a) The IT environment shall ensure that only secure values are accepted for security attributes.

FMT_MSA.3(a) Static attribute initialisation (secure management)

FMT_MSA.3.1(a) The IT environment shall enforce the [secure management SFP] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(a) The IT environment shall allow the [authorised administrators] to specify alternative initial values to override the default values when an object or information is created.

### 5.3.1 Administration server

For the purposes of this document, the Administration server is a required component of the TOE IT environment and is required to effectively realise the administrator role of the TOE.

The SFRs of the Administration server satisfy the assumption A.ADMINISTRATION_SERVER.

#### 5.3.1.1 FCS: Cryptographic Support

FCS_COP.1(d) Cryptographic operation (Administration server)

FCS_COP.1.1(d) The Administration server shall perform [encrypted and authenticated session based communication with the TOE] in accordance with a specified cryptographic algorithm [AES for encryption and DSA and SHA-1 for authentication] and cryptographic key sizes [256 bits for AES and 1024 bits for DSA] that meet the following: [FIPS 197 for AES and FIPS 186-2 for DSA and SHA-1].

#### 5.3.1.2 FIA: Identification and authentication

FIA_UAU.4(b) Single-use authentication mechanisms (Administration server)

FIA_UAU.4.1(b) The Administration server shall prevent reuse of authentication data related to [the secure management mechanism and the offline recovery mechanism].

#### 5.3.1.3 FMT: Security Management

FMT_MSA.1(b) Management of security attributes (Administration server)

FMT_MSA.1.1(b) The Administration server shall enforce the [secure management S FP] to restrict the ability to [assign, change_default, query, modify, delete]] the security attributes [all SafeBoot Device Encryption for PC Machine properties and User properties] to [authorised administrators].

FMT_MSA.2(b) Secure security attributes (Administration server)

FMT_MSA.2.1(b) The Administration server shall ensure that only secure values are accepted for security attributes.

FMT_MSA.3(b) Static attribute initialisation (Administration server)

FMT_MSA.3.1(b) The Administration server shall enforce the [secure management SFP] to provide [restrictive] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2(b) The Administration server shall allow the [authorised administrators] to specify alternative initial values to override the default values when an object or information is created.

FMT_MTD.1(e) Management of TSF data (audit)

FMT_MTD.1.1(e) The Administration server shall restrict the ability to [query, clear] the [TSF audit data] to [authorised administrators].

FMT_MTD.1(f) Management of TSF data (password)

FMT_MTD.1.1(f) The Administration server shall restrict the ability to [modify] the [a user's password] to [authorised administrators and a user may modify his own password if he successfully supplies his existing password first].

FMT_REV.1(b) Revocation (Administration server)

FMT_REV.1.1(b) The Administration server shall restrict the ability to revoke security attributes associated with the [users] within the TSC to [authorised administrators].

FMT_REV.1.2(b) The Administration server shall enforce the rules [Revocation can either take place the next time the user logs on, or the user can be revoked immediately, with their machine rebooted and their account disabled or deleted, as specified by the administrator].

FMT_SAE.1(b) Time-limited authorisation (Administration server)

FMT_SAE.1.1(b) The Administration server shall restrict the capability to specify an expiration time for [user passwords] to [authorised administrators].

FMT_SAE.1.2(b) For each of these security attributes, the Administration server shall be able to [give the user a warning that the password is about to expire a specified time before expiry, but also prevent the user from logging on until he has changed the password] after the expiration time for the indicated security attribute has passed.

FMT_SMR.1(b) Security roles (Administration server)

FMT_SMR.1.1(b) The Administration server shall maintain the roles [Administrator].

FMT_SMR.1.2(b) The Administration server shall be able to associate users with roles.

#### 5.3.1.4 FAU: Security Audit

FAU_SAR.1 Audit review

FAU_SAR.1.1 The Administration server shall provide [authorised administrators] with the capability to read [all audit information] from the audit records.

FAU_SAR.1.2 The Administration server shall provide the audit records in a manner suitable for the user to interpret the information.

FAU_SAR.3 Selectable audit review

FAU_SAR.3.1 The Administration server shall provide the ability to perform [sorting] of audit data based on [date and time, the event code, the object (machine or user) or the description of the audited event].

# 6 TOE Summary Specification

This section defines the instantiation of the security requirements for the TOE. This specification provides a description of the security functions and assurance measures of the TOE that meet the TOE security requirements.

## 6.1 TOE Security Functions

This section specifies the IT security functions of the TOE and how these functions satisfy the TOE security functional requirements. This includes a bi-directional mapping between functions and requirements that clearly shows which functions satisfy which requirements and that all requirements are met. Each security function contributes to the satisfaction of at least one TOE security functional requirement.

The table below maps these security functions to the security functional requirements already identified in section 5.1. Each function is given an identifier to enable unambiguous cross-referencing through the assurance documentation.

This section is presented in an informal style, and for ease of narrative and conciseness does not reproduce verbatim the text of the security functional requirements drawn from CC part 2 and mapped to the TOE in section 5.1 . If in some cases it is not obvious from the description that a security function embodies the security functional requirements expected from the table below, the mapping table ( Figure 11) should be taken as indicating that the requirement is included in the function, although for the purpose of the description in this document, that may be implicit.

| IT Security Function | TOE Security Functional Requirement |
|---|---|
| TSF.USER_ACCESS_CONTROL | FDP_ACC.2(a) Complete access control (user) |
| | FDP_ACF.1(a) Security attribute based access control (user) |
| | FTA_SSL.2 User-initiated locking |
| | FTA_TSE.1 TOE session establishment |
| | FPT_RVM.1 Non-bypassability of the TSP |
| TSF.USER_AUTHENTICATION | FTP_TRP.1 Trusted path |
| | FIA_UAU.2 User authentication before any action |
| | FIA_UAU.7 Protected authentication feedback |
| | FIA_UID.2 User identification before any action |
| | FIA_AFL.1 Authentication failure handling (user logon) |
| TSF.MANAGEMENT_BY_USER | FMT_MTD.1(b) Management of TSF data (password) |
| | FMT_SMF.1 Specification of Management Functions |
| TSF.HDD_ENCRYPTION | FCS_COP.1(a) Cryptographic operation (data encryption and decryption) |
| TSF.HDD_ENC_KEYMAN | FCS_CKM.1(a) Cryptographic key generation (symmetric) |
| | FCS_CKM.4 Cryptographic key destruction |
| | FCS_COP.1(b) Cryptographic operation (key encryption and decryption) |
| TSF.ADMIN_ACCESS_CONTROL | FPT_RVM.1 Non-bypassability of the TSP |
| TSF.SECURE_MANAGEMENT | FIA_UAU.4(a) Single-use authentication mechanisms (secure management)<br>FMT_MSA.1(a) Management of security attributes (secure management)<br>FCS_CKM.1(b) Cryptographic key generation (asymmetric)<br>FCS_CKM.4 Cryptographic key destruction<br>FCS_COP.1(c) Cryptographic operation (authenticated administration) |
| | FMT_MSA.1(a) Management of security attributes (secure management) |
| | FMT_MTD.1(a) Management of TSF data (audit) |
| | FMT_MTD.1(b) Management of TSF data (password) |
| | FMT_SMF.1 Specification of Management Functions |
| | FMT_MTD.2(a) Management of limits on TSF data (authentication failure) |
| | FMT_REV.1(a) Revocation |
| | FMT_SAE.1(a) Time-limited authorisation (secure management) |
| | FMT_SMR.1(a) Security roles |
| | FIA_ATD.1 User attribute definition |
| | FIA_UAU.4(a) Single-use authentication mechanisms (secure management) |
| TSF.SECURITY_AUDIT | FAU_GEN.1 Audit data generation |
| | FAU_GEN.2 User identity association |
| | FAU_STG.1 Protected audit trail storage |
| | FAU_STG.3 Action in case of possible audit data loss |
| TSF.PROTECTION | FPT_AMT.1 Abstract machine testing |
| | FPT_FLS.1 Failure with preservation of secure state |
| | FPT_RCV.1 Manual recovery |
| | FRU_FLT.1 Degraded fault tolerance |

**Figure 11 Mapping Security Functions to Security Functional Requirements**

### 6.1.1 User Access Control – TSF.USER_ACCESS_CONTROL

The SafeBoot Device Encryption Client replaces the master boot record on the bootable hard disk of the PC on which it is installed. So, when such a PC boots, the first code that gets loaded from the hard disk is the SafeBoot Device Encryption Client and the user is presented with the SafeBoot Device Encryption Client logon screen, and will be required to provide a valid user identifier and a valid, authenticated password before being granted access to the PC's data.

If a user attempts to bypass the SafeBoot Device Encryption Client logon by using a boot disk, for instance, they will be prevented from gaining access to the hard disk data by virtue of the fact that the hard disk is encrypted using AES and an encrypted key that cannot be subverted unless an attacker guesses the password or key or gains the password from a trusted individual.

Once a user has been granted access to the PC, they may choose to lock the PC, when leaving their desk for lunch, for instance, by activating the SafeBoot Device Encryption Client screen saver, which will then present the user with the SafeBoot Device Encryption Client logon screen when they try to use the PC again preventing unauthorised access.

A user whose account has been disabled will not be able to gain access to the TOE data, and if a user's account is disabled while he is using the TOE, he will be locked out and presented with the SafeBoot Device Encryption Client screen saver and not able to logon again until his account is enabled once again.

This functionality constitutes the machine access control SFP.

### 6.1.2 User Authentication – TSF.USER_AUTHENTICATION

When a user boots up a PC protected by SafeBoot Device Encryption for PC, they boot into the "SafeBoot Device Encryption Client OS", which is effectively what the TOE bootcode is, providing a trusted, secure and controlled environment in which the user may present his credentials (such as a user identity and a password) to the SafeBoot Device Encryption Client for authentication.

Before authentication can occur, the user must present the SafeBoot Device Encryption Client with his identity (as assigned).

When the user logs on, the credentials that he supplies are authenticated. In the case of a password, this is checked against a securely stored value associated with the user using the PKCS-5 algorithm (RFC 2898).

When a user presents his credentials to the SafeBoot Device Encryption Client for authentication at logon, his identity may be displayed in plain text, but for password authentication, there will only be feedback consisting of a '*' for each character typed while the authentication is in progress.

The user will be given a set number of opportunities to logon successfully, to cater for user error, but if the user exceeds the prescribed number of allowed consecutive failures, his account will be disabled.

This functionality contains the password authentication mechanism.

### 6.1.3 Management of TOE by User – TSF.MANAGEMENT_BY_USER

It is possible for a user to change his password as part of the logon process or from the SafeBoot Device Encryption Client screen saver, as long as they present their existing password for authentication as part of the process. This makes use of the password authentication mechanism

### 6.1.4 Hard Disk Encryption – TSF.HDD_ENCRYPTION

It is possible to subvert the logon process, for instance by using a bootable floppy disk, and so for this reason, the hard disk of the TOE is encrypted to prevent unauthorised user access to the TOE. This constitutes part of the machine access control SFP.

the SafeBoot Device Encryption Client operating system starts the crypt driver in memory once the user has entered the correct authentication information. From this point on the machine will look and behave as if the SafeBoot Device Encryption Client was not installed, with all hard disk access going through the

SafeBoot Device Encryption Client, such that data read from the hard disk is decrypted and data written to the hard disk is encrypted, using the hard disk encryption key of the TOE.

### 6.1.5 Hard Disk Encryption Key Management – TSF.HDD_ENC_KEYMAN

The TOE generates its hard disk encryption key in accordance with a FIPS 140-2 Compliant Key Generation Algorithm (a pseudo-random number generator based on DSS) with a key size of 256 bits.

The TSF destroys hard disk encryption keys by zeroing them when they are no longer in use, specifically when the TOE is uninstalled.

The hard disk encryption key is stored encrypted (using AES and a key length of 256 bits) under a key derived from the user's password. If the password changes, the hard disk encryption key is decrypted using the existing one and then encrypted for storage using the new password. The hard disk key itself does not change in such circumstances.

The hard disk key is decrypted as required when needed to access data on the TOE hard disk drive. This can only occur once a user has successfully logged on to the TOE.

### 6.1.6 Administrative Access Control – TSF.ADMIN_ACCESS_CONTROL

Management of SafeBoot Device Encryption for PC TOEs is via the administration secure management interface. Any administrator wishing to manage a SafeBoot Device Encryption for PC TOE must first establish a secure management session with that TOE.

Once a secure session has been established (using an authenticated message exchange), the administrators ability to modify TOE attributes is governed by the privilege level of the administrator in respect to those attributes in relation to that TOE machine.

This function constitutes the secure management access control SFP.

### 6.1.7 Secure Management – TSF.SECURE_MANAGEMENT

The user may change his own password, however the bulk of the management of the TOE functionality must be performed by an administrator. All administrator configuration options relevant to the TOE machine and its users are detailed in section10. This section discusses various aspects of secure management and some of the key configuration options in more detail.

The client and administrator create a secure session for allowing secure configuration to occur. This mutual authentication is performed using DSA signatures, and the session is then established using AES encryption of all link traffic using keys generated using Diffie-Hellman key. This is a single use authentication mechanism, it is not possible to create a new session by replaying old authentication data. No administration is permitted before an authenticated administration session has been established.

Only an authorised administrator may add, modify or delete users and their attributes on the TOE. For instance, set or reset a user's password, and may also configure the password policy associated with that user, such as enforcing password length or content, password history, or lifetime. Also, the number of times that logon failure may consecutively occur before the user account is disabled may be set by the administrator.

For each user, the administrator creates a User Identitifier, and defines a password policy. During installation a hard disk encryption key is generated along with user encryption keys that are stored in the SafeBoot Device Encryption for PC object database.

An authorised administrator can view and if required clear the audit data from a TOE.

If a user is deleted, his details will be removed from the object database once the TOE is resynchronised.

When the administrative session is terminated, all of the session keys that were created are then zeroed.

In addition, during an authorised administrative session, the administrator may:

- Synchronise the TOE with the object directory to invoke any configuration changes,

- Reboot the TOE,
- Lock the TOE to the screen saver,
- Disable the account of the current user,
- Deploy files to the TOE,
- Recover the TOE in the event of a lost password

This function constitutes the secure management SFP and contains the secure management mechanism.

### 6.1.8 Audit – TSF.SECURITY_AUDIT

The TOE maintains an audit log. This contains a list of events that have occurred on the TOE, and each entry consists of a timestamp, type of event, user ID of the user logged on at the time and the result of the event. The audit functions are always active while the TOE is operational.

All of the following result in audit entries:

All try events, resulting from:
- Attempts to make changes to passwords
- Recovery attempts
- Expiry and timeouts

All success events, such as
- Changes to passwords
- Logon
- Recovery

All failure events.
- Password change failures
- Logon failures
- Recovery failures

The audit log can only hold 3000 entries. When it is full, each new entry added results in the oldest entry in the log becoming overwritten.

The audit log can only be viewed or cleared by authorised administrators, and he can choose to view the entries ordered on a number of factors, specifically: date and time, the event code, the object (machine or user) or the description of the audited event.

As the hard disk is encrypted (the TOE operates in CC mode), access to the audit trail of the TOE is restricted, and protected from unauthorised modification or deletion.

### 6.1.9 Self-Protection of the TOE – TSF.PROTECTION

The TOE has a number of related functions that help to maintain its integrity under certain circumstances, such as hardware failure, or communications link failure.

The TSF runs a suite of tests during initial start-up, and in the case of the random number generator test, continuously to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF. These tests are described in sections 2.10 and 2.11.

The TSF preserves a secure state when communications with the administration server are unexpectedly terminated or when there is a power failure to the TOE.

After a user account has been disabled or the user has forgotten their logon password when they try to logon, the TSF enters a maintenance mode where the ability to recover the normal functionality of the TOE is provided either online via a secure administration session, or offline using the offline recovery procedure (This involved providing security credentials to a human administrator for manual authentication to a central site in order to obtain a recovery code to reactivate the TOE).

The TSF ensures that normal operation continues when the link to the administration server is lost, by maintaining a local copy of the object database. The TOE can be configured such that if this link is lost and the TOE remains unsynchronised for a specified amount of time, then the TOE is locked.

## 6.2 Assurance Measures

The requirements of the assurance components that the TOE must satisfy in order to achieve certification at EAL4 are each addressed by a document specifically written for the purpose. The following table names each of these assurance documents.

| Assurance components | Assurance Measures |
|---|---|
| ACM_AUT.1 Partial CM automation | SafeBoot 5 Configuration Management Deliverables Package |
| ACM_CAP.4 Generation support and acceptance procedures | SafeBoot 5 Configuration Management Deliverables Package |
| ACM_SCP.2 Problem tracking CM coverage | SafeBoot 5 Configuration Management Deliverables Package |
| ADO_DEL.2 Detection of modification | SafeBoot 5 Delivery and Operation Deliverables Package |
| ADO_IGS.1 Installation, generation, and start-up procedures | SafeBoot 5 Delivery and Operation Deliverables Package |
| ADV_FSP.2 Fully defined external interfaces | SafeBoot 5 Functional Specification |
| ADV_HLD.2 Security enforcing high -level design | SafeBoot 5 High-Level Design |
| ADV_IMP.1 Subset of the implementation of the TSF | SafeBoot 5 Low -Level Design Deliverables Package |
| ADV_LLD.1 Descriptive low-level design | SafeBoot 5 Low -Level Design Deliverables Package |
| ADV_RCR.1 Informal correspondence demonstration | SafeBoot 5 Low -Level Design Deliverables Package |
| ADV_SPM.1 Informal TOE security policy model | SafeBoot 5 Low -Level Design Deliverables Package |
| AGD_ADM.1 Administrator guidance | SafeBoot Administrators Guide |
| AGD_USR.1 User guidance | SafeBoot Administrators Guide |
| ALC_DVS.1 Identification of security measures | SafeBoot 5 Life Cycle Deliverables Package |
| ALC_LCD.1 Developer defined life-cycle model | SafeBoot 5 Life Cycle Deliverables Package |
| ALC_TAT.1 Well-defined development tools | SafeBoot 5 Life Cycle Deliverables Package |
| ATE_COV.2 Analysis of coverage | SafeBoot 5 Testing Deliverables Package |
| ATE_DPT.1 Testing: high -level design | SafeBoot 5 Testing Deliverables Package |
| ATE_FUN.1 Functional testing | SafeBoot 5 Testing Deliverables Package |
| ATE_IND.2 Independent testing - sample | SafeBoot 5 Testing Deliverables Package |
| AVA_MSU.2 Validation of analysis | SafeBoot 5 Vulnerability Assessment Deliverables Package |
| AVA_SOF.1 Strength of TOE security function evaluation | SafeBoot 5 Vulnerability Assessment Deliverables Package |
| AVA_VLA.2 Independent vulnerability analysis | SafeBoot 5 Vulnerability Assessment Deliverables Package |

**Figure 12 Mapping of Assurance Components to Assurance Measures**

# 7 Administration Server Summary Specification

This section defines the instantiation of the security requirements for the Administration Server. This specification provides a description of the security functions and assurance measures of the Administration Server that meet the Administration Server security requirements as defined in section 5.3.1.

## 7.1 Administration Server Security Functions

This section specifies the IT security functions of the Administration Server and how these functions satisfy the Administration Server security functional requirements. This includes a bi-directional mapping between functions and requirements that clearly shows which functions satisfy which requirements and that all requirements are met. Each security function contributes to the satisfaction of at least one Administration Server security functional requirement.

The table below maps these security functions to the security functional requirements already identified in section 5.3. Each function is given an identifier to enable unambiguous cross-referencing through the assurance documentation.

This section is presented in an informal style, and for ease of narrative and conciseness does not reproduce verbatim the text of the security functional requirements drawn from CC part 2 and mapped to the Administration Server in section 5.3. If in some cases it is not obvious from the description that a security function embodies the security functional requirements expected from the table below, the mapping table (Figure 15) should be taken as indicating that the requirement is included in the function, although for the purpose of the description in this document, that may be implicit.

| IT Security Function | Administration Server Security Functional Requirement |
|---|---|
| TSF.ADMINISTRATION_SERVER | FCS_COP.1(d) Cryptographic operation (Administration server)<br>FIA_UAU.4(b) Single-use authentication mechanisms (Administration server)<br>FMT_MSA.1(b) Management of security attributes (Administration server) |
| | FMT_MSA.2(b) Secure security attributes (Administration server) |
| | FMT_MSA.3(b) Static attribute initialisation (Administration server) |
| | FMT_MTD.1(e) Management of TSF data (audit) |
| | FMT_MTD.1(f) Management of TSF data (password) |
| | FMT_REV.1(b) Revocation (Administration server) |
| | FMT_SAE.1(b) Time-limited authorisation (Administration server) |
| | FMT_SMR.1(b) Security roles (Administration server) |
| | FAU_SAR.1 Audit Review |
| | FAU_S AR.3 Selectable Audit Review |

**Figure 13 Mapping Security Functions to Security Functional Requirements**

### 7.1.1 Administration Server – TSF.ADMINISTRATION_SERVER

This function gives an authorised administrator access to a GUI that allows him to configure and manage the TOE. TSF.ADMINISTRATION_SERVER allows the administration server side of the functions TSF.ADMIN_ACCESS_CONTROL, TSF.SECURE_MANAGEMENT and TSF.SECURITY_AUDIT.

It also provides a user interface through which an authorised administrator may view or selectively review audit data from the TOE.

# 8 Protection Profile Claims

This Security Target does not include any claims that the TOE conforms to any named Protection Profile.

# 9 Rationale

The purpose of this section is to demonstrate that the threats, assumptions and organisational security policies identified in the TOE security environment (see section 3) are satisfied by the security objectives described in section 4. Further, this section also demonstrates that these objectives are satisfied by the security functional requirements identified in section 5.

## 9.1 Security Objectives Rationale

The following table demonstrates that each threat identified in the TOE security environment is countered by one or more security objectives. Conversely, each security objective (either solely or in collection with other objectives) matches at least one assumption, threat or procedure.

| Threats, assumptions and organisational security policies | Corresponding security objectives |
|---|---|
| A.MANAGEMENT | OE.MANAGED |
| A.NO_MALEVOLENCE | OE.MANAGED |
| A.PROFICIENT_USERS | OE.EASE_OF_USE_ADMIN<br>OE.EASE_OF_USE_USER |
| A.AUTHENTICATION_DATA_PRIVATE | OE.EASE_OF_USE_USER<br>OE.AUTH |

| Threats, assumptions and organisational security policies | Corresponding security objectives |
|---|---|
| A.TIME_SOURCE | OE.TIME_SOURCE |
| A.ADMINISTRATION_SERVER | OE.ADMINISTRATION_SERVER |
| A.SECURE_BACKUP | OE.SECURE_BACKUP |
| A.AVAILABLE_BACKUP | OE.AVAILABLE_BACKUP |
| A.DOMAIN_SEPARATION | OE.DOMAIN_SEPARATION |
| A.TRUSTED_SOFTWARE | OE.TRUSTED_SOFTWARE |
| A.NON_TECHNICAL_IDENTITY_VERIFICATION | OE.NON_TECHNICAL_IDENTITY_VERIFICATION |
| T.ACCESS | O.AUTHORISATION<br>O.ACCESS_CONTROL<br>O.AUDIT |
| T.ALTERNATE_BOOT_PROCESS | O.ENCRYPTED_MEDIA |
| T.CONFIG _MODIFICATION | O.PROTECT |
| T. CORRUPT_AUDIT | O.AUTHORISATION<br>O.ENCRYPTED_MEDIA<br>O.EFFECTIVE_ADMINISTRATION<br>O.AUDIT |
| T.EASE_OF_USE_ADMIN | OE.EASE_OF_USE_ADMIN |
| T.EASE_OF_USE_USER | O.EASE_OF_USE_USER |
| T.EAVESDROP_TRANSIT | O.DATA_TRANSFER |
| T.OBJECT_REUSE | O.NO_OBJECT_REUSE |
| T.PASSWORD_LOSS | O.SECURE_RECOVERY<br>OE.NON_TECHNICAL_IDENTITY_VERIFICATION |
| T.RECORD_ACTIONS | O.AUTHORISATION<br>O.AUDIT |
| T.RECOVERY_PROCEDURE_INTERCEPT | O.NO_OBJECT_REUSE |
| T.RECOVERY_MASQUERADE | O.AUTHORISATION<br>OE.NON_TECHNICAL_IDENTITY_VERIFICATION |
| T.REMOVE_DISK | O.ENCRYPTED_MEDIA |
| T.SPOOF<br><br>T.SYSTEM_ACCESS | O.TRUSTED_PATH<br>O.DATA_TRANSFER<br>O.AUTHORISATION<br>O.PROTECT<br>OE.MANAGED |
| T.UNAUTHORISED_MODIFICATION | O.AUTHORISATION<br>O.ENCRYPTED_MEDIA<br>O.PROTECT<br>OE.MANAGED<br>OE.EASE_OF_USE_USER |
| P.AUTHORISED_USERS | O.AUTHORISATION |
| P.CRYPTOGRAPHIC_KEYS | O.CRYPOTOGRAPHIC_KEYS |
| P.CRYPTOGRAPHIC_OPERATIONS | O.CRYPTOGRAPHIC_OPERATIONS |
| P.EAVESDROP_TRANSIT | O.DATA_TRANSFER |
| P.FAULT_TOLERANCE | O.FAULT_TOLERANCE |
| P.USER_ACCOUNTABILITY | OE.AUTH<br>O.AUDIT |

**Figure 14 Mapping Threats, Assumptions and Policies to Objectives**

| Security Objectives | Corresponding threats, assumptions and organisational security policies |
|---|---|
| OE.MANAGED | A.MANAGEMENT<br>A.NO_MALEVOLENCE<br>T.SYSTEM_ACCESS<br>T.UNAUTHORISED_MODIFICATION |
| OE.EASE_OF_USE_ADMIN | A.PROFICIENT_USERS<br>T.EASE_OF_USE_ADMIN |
| OE.EASE_OF_USE_USER | A.PROFICIENT_USERS |

| Security Objectives | Corresponding threats, assumptions and organisational security policies |
|---|---|
| | A.AUTHENTICATION_DATA_PRIVATE<br>T.UNAUTHORISED_MODIFICATION |
| OE.AUTH | A.AUTHENTICATION_DATA_PRIVATE<br>P.USER_ACCOUNTABILITY |
| OE.TIME_SOURCE | A.TIME_SOURCE |
| OE.ADMINISTRATION_SERVER | A.ADMINISTRATION_SERVER |
| OE.SECURE_BACKUP | A.SECURE_BACKUP |
| OE.AVAILABLE_BACKUP | A.AVAILABLE_BACKUP |
| OE.DOMAIN_SEPARATION | A.DOMAIN_SEPARATION |
| OE.TRUSTED_SOFTWARE | A.TRUSTED_SOFTWARE |
| OE.NON_TECHNICAL_IDENTITY_VERIFICATION | A.NON_TECHNICAL_IDENTITY_VERIFICATION<br>T.PASSWORD_LOSS<br>T.RECOVERY_MASQUERADE |
| O.AUTHORISATION<br><br><br><br><br><br><br>O.ACCESS_CONTROL<br>O.AUDIT | T.ACCESS<br>T. CORRUPT_AUDIT<br>T.RECORD_ACTIONS<br>T.RECOVERY_MASQUERADE<br>T.SYSTEM_ACCESS<br>T.UNAUTHORISED_MODIFICATION<br>P.AUTHORISED_USERS<br>T.ACCESS<br>T.ACCESS<br>T. CORRUPT_AUDIT<br>T.RECORD_ACTIONS<br>P.USER_ACCOUNTABILITY |
| O.ENCRYPTED_MEDIA | T.ALTERNATE_BOOT_PROCESS<br>T.CORRUPT_AUDIT<br>T.UNAUTHORISED_MODIFICATION<br>T.REMOVE_DISK |
| O.PROTECT | T.CONFIG _MODIFICATION<br>T.SYSTEM_ACCESS<br>T.UNAUTHORISED_MODIFICATION |
| O.EFFECTIVE_ADMINISTRATION | T.CORRUPT_AUDIT |
| O.EASE_OF_USE_USER | T.EASE_OF_USE_USER |
| O.DATA_TRANSFER | T.EAVESDROP_TRANSIT<br>P.EAVESDROP_TRANSIT |
| O.NO_OBJECT_REUSE | T.OBJECT_REUSE<br>T.RECOVERY_PROCEDURE_INTERCEPT |
| O.SECURE_RECOVERY | T.PASSWORD_LOSS |
| O.TRUSTED_PATH | T.SPOOF |
| O.CRYPOTOGRAPHIC_KEYS | P.CRYPTOGRAPHIC_KEYS |
| O.CRYPTOGRAPHIC_OPERATIONS | P.CRYPTOGRAPHIC_OPERATIONS |
| O.FAULT_TOLERANCE | P.FAULT_TOLERANCE |

**Figure 15 Mapping Security Objectives to Threats, Assumptions and Policies**

#### 9.1.1.1 OE.MANAGED

Those responsible for the TOE ensure that it is managed securely. Specifically, one or more competent individuals are assigned management responsibility for the TOE (A.MANAGEMENT). These individuals are expected to behave professionally and are trusted to behave in a way that maintains the security of the TOE (A.NO_MALEVOLENCE). If the TSF is configured securely and its users and administrators act in accordance with their training in its correct use, then, if all other TSF security objectives are met, there should be no way for an unauthorized user to gain access to or modify the TOE, thus countering the threats T.SYSTEM_ACCESS and T.UNAUTHORISED_MODIFICATION.

#### 9.1.1.2 OE.EASE_OF_USE_ADMIN

The TOE is managed by proficient administrators that have been trained in its use and follow the guidance laid down for its secure use (A.PROFICIENT_USERS). One measure of this proficiency is that administrators check their actions to ensure that they do not inadvertently configure the TSF in an insecure fashion, countering the threat T.EASE_OF_USE_ADMIN..

#### 9.1.1.3 OE.EASE_OF_USE_USER

The TOE is used by proficient users that have been trained in its use and follow the guidance laid down for its secure use (A.PROFICIENT_USERS). Specifically, users are expected to not leave the TOE unattended in a logged in state, ensuring that it cannot be modified and so countering the threat T.UNAUTHORISED_MODIFICATION. Users are expected to keep their secure user credentials secret and so counter the threat T.AUTHENTICATION_DATA_PRIVATE.

#### 9.1.1.4 OE.AUTH

Users and administrators of the TOE are expected to keep their secure user credentials secret, and so counter the threat T.AUTHENTICATION_DATA_PRIVATE. As an incentive, users may be held accountable for all security relevant actions carried out on the TSF (P.USER_ACCOUNTABILITY), and all such actions are audited, although this is covered by a separate objective, O.AUDIT.

#### 9.1.1.5 OE.TIME_SOURCE

The IT environment provides a reliable source of time information to enable the TSF to timestamp its audit records (A.TIME_SOURCE).

#### 9.1.1.6 OE.ADMINISTRATION_SERVER

The IT environment for the TOE contains an administration server that provides all of the functions to manage the TSF (A.ADMINSTRATIVE_SERVER).

#### 9.1.1.7 OE.SECURE_BACKUP

User's data backups are separately encrypted or physically protected to ensure data security is not compromised through theft of or unauthorised access to backup information (satisfying the assumption A.SECURE_BACKUP).

#### 9.1.1.8 OE.AVAILABLE_BACKUP

Regular and complete backups are taken to enable recovery of user data in the event of loss or damage to data as a result of the actions of a threat agent (A.AVAILABLE_BACKUP).

#### 9.1.1.9 OE.DOMAIN_SEPARATION

Separate threads of execution for TOE processes enable the TOE to be protected from potential attack from malicious software processes (A.DOMAIN_SEPARATION).

#### 9.1.1.10 OE.TRUSTED_SOFTWARE

Running only trusted software in the TOE IT environment and taking other relevant measures such as using anti-virus software and firewalls, etc. as appropriate protects the TOE against attack from malicious software and enables it to target its specific threat profile (A.TRUSTED_SOFTWARE).

#### 9.1.1.11 OE.NON_TECHNICAL_IDENTITY_VERIFICATION

This objective, that there is a database of authorised TSF-users along with user-specific authentication data for the purpose of enabling administrative personnel to verify the identity of a user over a voice-only telephone line before providing them with support directly addresses the assumption A.NON_TECHNICAL_IDENTITY_VERIFICATION. Ordinarily, recovery would take place using a secure management session, but there are times when this is not possible, such as when the use has no network connection to the administration server. By allowing a user to be authenticated by non-technical means, it allows the administrator to reset the user's password in the event of password loss, thus

countering the threat T.PASSWORD_LOSS. By providing a mechanism for the non-technical verification of the identity of a user, this objective counters the threat T.RECOVERY_MASQUERADE. There is a threat that this recovery mechanism can be subverted through an attacker overhearing the recovery process and impersonating the user with the authentication information. This threat is addressed by the objective O.AUTHORISATION.

#### 9.1.1.12 O.AUTHORISATION

This objective is at the heart of what SafeBoot Device Encryption for PC does. SafeBoot Device Encryption for PC provides access control and does not allow any user access until their credentials have been authenticated and so addresses the threats T.ACCESS, T.SYSTEM_ACCESS, T.RECORD_ACTIONS, T.UNAUTHORISED_MODIFICATION and T.CORRUPT_AUDIT. By implementing access control with authentication, this objective implements the policy P.AUTHORISED_USERS.

If a TOE is stolen, this fact is used to allow it to be disabled by the administration server. If the machine is connected to the administration server, it can be disabled so that no user can logon to it. If it is not connected to the administration server, and the thief tries to gain access to it via the offline recovery mechanism, he will be denied, even though he may be able to convincingly masquerade as a genuine user. This addresses the threat T.RECOVERY_MASQUERADE.

#### 9.1.1.13 O.ACCESS_CONTROL

The TSF provides access control. This objective along with O.AUTHORISATION and O.AUDIT counters the threat T.ACCESS

#### 9.1.1.14 O.AUDIT

The TSF audits certain events to allow authorized administrators to monitor how the TOE is being used and potentially to detect any attempts to undermine its security. O.AUDIT implements the policy P.USER_ACCOUNTABILITY. By recording audit events and by requiring administrators to be authenticated before being able to view or clear audit information, this objective is partly responsible for countering the threats T.ACCESS, T.CORRUPT_AUDIT and T.RECORD_ACTIONS.

#### 9.1.1.15 O.ENCRYPTED_MEDIA

Along with access control, the TSF encrypts its hard disk so that any attempts to bypass access control will fail as the attacker will only have gained access to encrypted data that he will not be able to decrypt without also obtaining the hard disk encryption key. This objective therefore counters the threats T.ALTERNATE_BOOT_PROCESS and T.REMOVE_DISK. By encrypting the hard disk, this also protects the audit trail and any other data or applications stored on the hard disk against unauthorized modification, thus countering T.UNAUTHORISED_MODIFICATION and T.CORRUPT_AUDIT.

#### 9.1.1.16 O.PROTECT

The TSF provides synchronization and self-test facilities to help it to detect any unauthorized modification or accidental corruption of its own configuration or resources. This counters the threat T.CONFIG_MODIFICATION, T.SYSTEM_ACCESS and T.UNAUTHORISED_MODIFICATION

#### 9.1.1.17 O.EFFECTIVE_ADMINISTRATION

This is in some ways an objective that is made up of aspects of other objectives (O.AUTHORISATION, OE.ADMINISTRATION_SERVER, OE.MANAGED, OE.EASE_OF_USE_ADMIN, O.DATA_TRANSFER, O.AUDIT) and is included to emphasise the importance of administration to the TOE.

#### 9.1.1.18 O.EASE_OF_USE_USER

The only function that the user may perform at the client interface that affects the configuration of security is the ability to change his password. The administrator defines the password policy such that the user is not able to change his password to a value that contravenes this password policy. This objective thus counters the threat T.EASE_OF_USE_USER.

#### 9.1.1.19 O.DATA_TRANSFER

This objective implements the policy P.EAVESDROP_TRANSIT using DSS and AES block encryption to authenticate and encrypt all transmissions between the TSF and the administration server. By doing so, it prevents unauthorized access to the information in the transmissions, thus countering the threat T.EAVESDROP_TRANSIT. The protocol for establishing a secure management session is a one-time authentication protocol, preventing an attacker from establishing a secure management session by replaying transmissions that he has recorded previously, and so countering the threat.

#### 9.1.1.20 O.NO_OBJECT_REUSE

The TSF uses one-time authentication mechanisms for both secure management and offline recovery. This counters the threats T.OBJECT_REUSE and T.RECOVERY_PROCEDURE_INTERCEPT. T.OBJECT_REUSE could also be a threat if the password mechanism allowed expired passwords to persist, but this is not the case. When a password is changed, the password that it replaces is no longer valid and cannot be used to gain access to the information stored on the TOE.

#### 9.1.1.21 O.SECURE_RECOVERY

If a user forgets his password then there is the possibility that the TSF protected information will be lost. This objective counters the threat of T.PASSWORD_LOSS by providing a secure recovery mechanism to allow the user to regain authenticated access to the TOE using a new password.

#### 9.1.1.22 O.TRUSTED_PATH

This objective counters the threat that an attacker may impersonate the TSF in an attempt to gain the user's logon credentials. This objective along with others (O.UNATHORISED_MODIFICATION and O.AUTHORISED) counters this threat by providing a secure and reliable link between the user and the TSF.

#### 9.1.1.23 O.CRYPOTOGRAPHIC_KEYS

The TSF ensures that cryptographic keys are generated, accessed, protected, and destroyed in accordance with requirements defined by FIPS 140-2 Level 1. O.CRYPTOGRAPHIC_KEYS implements the security policy P.CRYPTOGRAPHIC_KEYS.

#### 9.1.1.24 O.CRYPTOGRAPHIC_OPERATIONS

The TSF must ensure that all cryptographic operations used to protect information and encryption keys are compliant with the standards defined by FIPS 140-2 Level 1 and FIPS 197 (AES). O.CRYPTOGRAPHIC_OPERATIONS implements the security policy P.CRYPTOGRAPHIC_OPERATIONS.

#### 9.1.1.25 O.FAULT_TOLERANCE

This objective implements the policy P.FAULT_TOLERANCE.

## 9.2 Security Requirements Rationale

The Security Requirements for the TOE have been chosen to meet its Security Objectives effectively. All Functional Requirements have been selected directly from CC part 2 (where a requirement is dependent on one or more other SFRs, all dependencies have been selected), and all Assurance Requirements directly from CC part 3.

The figure below demonstrates that each TOE Security Objective is satisfied by one or more SFRs.

| Security Objectives | TOE Security Functional Requirement |
|---|---|
| O.ACCESS_CONTROL | FDP_ACC.2(a) Complete access control (user)<br>FDP_ACF.1(a) Security attribute based access control (user)<br>FTA_SSL.2 User-initiated locking<br>FTA_TSE.1 TOE session establishment<br>FPT_RVM.1 Non-bypassability of the TSP |
| O.FAULT_TOLERANCE | FRU_FLT.1 Degraded fault tolerance |

| Security Objectives | TOE Security Functional Requirement |
|---|---|
| O.TRUSTED_PATH | FTP_TRP.1 Trusted path |
| O.AUTHORISATION | FIA_UAU.2 User authentication before any action<br>FIA_UAU.7 Protected authentication feedback<br>FIA_UID.2 User identification before any action<br>FIA_AFL.1 Authentication failure handling (user logon) |
| O.CRYPTOGRAPHIC_OPERATIONS | FCS_COP.1(a) Cryptographic operation (data encryption and decryption)<br>FCS_COP.1(b) Cryptographic operation (key encryption and decryption)<br>FCS_COP.1(c) Cryptographic operation (authenticated administration) |
| O.CRYPTOGRAPHIC_KEYS | FCS_CKM.1(a) Cryptographic key generation (symmetric)<br>FCS_CKM.4 Cryptographic key destruction |
| O.ENCRYPTED_MEDIA | FCS_COP.1(a) Cryptographic operation (data encryption and decryption)<br>FIA_ATD.1 User attribute definition |
| O.NO_OBJECT_REUSE | FIA_UAU.4(a) Single-use authentication mechanisms (secure management) |
| O.EASE_OF_USE_USER | FMT_MTD.1(b) Management of TSF data (password)<br>FMT_SMF.1 Specification of Management Functions<br>FMT_SMR.1(a) Security roles |
| O.EFFECTIVE_ADMINISTRATION and OE.ADMINISTRATION_SERVER | FPT_RVM.1 Non-bypassability of the TSP<br>FCS_CKM.1(b) Cryptographic key generation (asymmetric)<br>FCS_CKM.4 Cryptographic key destruction<br>FMT_MSA.1(a) Management of security attributes (secure management)<br>FMT_MSA.2(a) Secure security attributes (secure management)<br>FMT_MSA.3(a) Static attribute initialisation (secure management)<br>FMT_MTD.1(a) Management of TSF data (audit)<br>FMT_MTD.1(b) Management of TSF data (password)<br>FMT_MTD.2(a) Management of limits on TSF data (authentication failure)<br>FMT_SAE.1(a) Time-limited authorisation (secure management)<br>FMT_REV.1(a) Revocation<br>FMT_SMR.1(a) Security roles<br>FIA_UAU.4(a) Single-use authentication mechanisms (secure management)<br>FIA_UAU.7 Protected authentication feedback |
| O.DATA_TRANSFER | FPT_RVM.1 Non-bypassability of the TSP<br>FCS_CKM.1(b) Cryptographic key generation (asymmetric)<br>FCS_CKM.4 Cryptographic key destruction<br>FIA_UAU.4(a) Single-use authentication mechanisms (secure management)<br>FIA_UAU.7 Protected authentication feedback |
| O.SECURE_RECOVERY | FIA_UAU.4(a) Single-use authentication mechanisms (secure management) |
| O.AUDIT | FAU_GEN.1 Audit data generation<br>FAU_GEN.2 User identity association<br>FAU_STG.1 Protected audit trail storage<br>FAU_STG.3 Action in case of possible audit data loss<br>FAU_SAR.1 Audit review<br>FAU_SAR.3 Selectable audit review |
| OE.TIME_SOURCE | FAU_GEN.1 Audit data generation |
| O.PROTECT | FPT_AMT.1 Abstract machine testing<br>FPT_FLS.1 Failure with preservation of secure state |

| Security Objectives | TOE Security Functional Requirement |
|---|---|
| | FPT_RCV.1 Manual recovery<br>FPT_SEP.1 TSF domain separation |
| OE.SECURE_BACKUP | AGD_ADM.1 Administrator Guidance |
| OE.AVAILABLE_BACKUP | AGD_ADM.1 Administrator Guidance |
| OE.MANAGED | AGD_ADM.1 Administrator Guidance |
| OE.EASE_OF_USE_USER | AGD_ADM.1 Administrator Guidance |
| OE.AUTH | AGD_USR.1 User Guidance |
| OE.EASE_OF_USE_ADMIN | AGD_USR.1 User Guidance |

**Figure 16 Mapping of Security Objectives to Functional and Assurance Requirements**

| Security Objectives | TOE Security Functional Requirement | Rationale |
|---|---|---|
| O.ACCESS_CONTROL | FDP_ACC.2(a) | O.ACCESS_CONTROL is the access control policy objective. This maps directly onto FDP.ACC.2(a). |
| | FDP_ACF.1(a) | FDP_ACF.1(a) states that the access control policy should be implemented with respect to the user's identity and password |
| | FTA_SSL.2<br><br><br>FTA_TSE.1 | FTA_SSL.2 covers user-initiated locking of the TOE, invoking access control before any user can regain access to the TOE.<br>Part of the access control policy is that user accounts may be disabled and users thereby denied access. This is provided by FTA_TSE.1. |
| | FPT_RVM.1 | FPT_RVM.1 ensures that access control is always applied before a user can gain authorised access to the TOE |
| O.FAULT_TOLERANCE | FRU_FLT.1 | FRU_FLT.1 stipulates that normal operation of the TSF shall continue when communication is lost with the administration server. This matches the objective O.FAULT_TOLERANCE. |
| O.TRUSTED_PATH | FTP_TRP.1 | There is a trusted path to enable users to be confident of the security of the link between the user and the TSF. FTP_TRP.1 matches O.TRUSTED_PATH. |
| O.AUTHORISATION | FIA_UAU.2 | O.AUTHORISATION ensures that users are authenticated before they are permitted access to the TOE. FIA_UAU.2 stipulates that users are required to be authenticated before they can use the TOE. |
| | FIA_UAU.7 | FIA_UAU.7 describes the user feedback given during user authentication. |
| | FIA_UID.2 | As part of the authentication process, users must be identified. This functionality is described in FIA_UID.2. |
| | FIA_AFL.1 | FIA_AFL.1 describes how user authentication failure is handled. |
| O.CRYPTOGRAPHIC_OPERATIONS | FCS_COP.1(a) | FCS_COP.1(a) states that AES is used for data encryption and decryption. This is FIPS 140-2 compliant. |
| | FCS_COP.1(b) | FCS_COP.1(b) states that AES shall be used for key encryption by the TSF. This is FIPS 140-2 compliant. |
| | FCS_COP.1(c) | FCS_COP.1(c) states that DSS is used to establish secure management communications and AES is used for encrypting communication data blocks. This is FIPS 140-2 compliant. |

| Security Objectives | TOE Security Functional Requirement | Rationale |
|---|---|---|
| O.CRYPTOGRAPHIC_KEYS | FCS_CKM.1(a) | O.CRYPTOGRAPHIC_KEYS states that cryptographic keys are generated, accessed, protected, and destroyed in accordance with requirements defined by FIPS 140-2. FCS_CKM.1(a) states that keys will be generated using mechanisms that are compliant with FIPS 140-2. |
| | FCS_CKM.4 | FCS_CKM.4 states that keys will be destroyed using mechanisms that are compliant with FIPS 140-2. |
| O.ENCRYPTED_MEDIA | FCS_COP.1(a) | FCS_COP.1(a) states that data is encrypted and decrypted using AES, providing the encrypted media called for by O.ENCRYPTED_MEDIA. |
| | FIA_ATD.1 | A number of user attributes are stored securely on the TSF. |
| O.NO_OBJECT_REUSE | FIA_UAU.4(a) | FIA_UAU.4(a) stipulates that the single-use authentication mechanisms (offline recovery and secure management session establishment) operate in a way that prevents reuse of authentication data as a means of attack. This satisfies the objective O.NO_OBJECT_REUSE |
| O.EASE_OF_USE_USER | FMT_MTD.1(b) | O.EASE_OF_USE_USER stipulates that an authorised user should be permitted to change his password. FMT_MTD.1(b) satisfies that objective. |
| | FMT_SMF.1 | FMT_SMF.1 specifies that the only security management function available to an authorised user is to change his password. |
| | FMT_SMR.1(a) | FMT_SMR.1(a) specifies that there is a user role. |
| O.EFFECTIVE_ADMINISTRATION AND OE.ADMINISTRATION_SERVER | FPT_RVM.1 | FPT_RVM.1 stipulates that access control shall always operate and that there will be no other way of gaining access to the TSF. |
| | FCS_CKM.1(b) | FCS_CKM.1(b) specifies the means by which the key material used to secure the management of the TSF is generated. |
| | FCS_CKM.4 | FCS_CKM.4 specifies how these keys are to be destroyed once they are no longer required |
| | FMT_MSA.1(a) | FMT_MSA.1(a) specifies that secure management is realised by implementing the secure management SFP. |
| | FMT_MSA.2(a) | FMT_MSA.2(a) stipulates that only secure attributes can be used to configure the TSF. |
| | FMT_MSA.3(a) | FMT_MSA.3(a) specifies that default values may be assigned by the secure management SFP and that these may be overridden by an authorised administrator to provide alternative values. |
| | FMT_MTD.1(a) | FMT_MTD.1(a) specifies that authorised administrators may view or clear the TSF audit data. |
| | FMT_MTD.1(b) | FMT_MTD.1(a) specifies that authorised administrators may set user passwords for the TSF. |
| | FMT_MTD.2(a) | FMT_MTD.2(a) allows authorised administrators to determine how repeated user logon failures are to be handled |
| | FMT_SAE.1(a) | FMT_SAE.1(a) allows only administrators to set the lifetime of a user password |
| | FMT_REV.1(a) | FMT_REV.1(a) allows authorised administrators to |

| Security Objectives | TOE Security Functional Requirement | Rationale |
|---|---|---|
| | | revoke user security attributes, that is disable user accounts. |
| | FMT_SMR.1(a) | FMT_SMR.1(a) specifies that there is an administrator role. |
| | FIA_UAU.4(a) | FIA_UAU.4(a) stipulates that the single-use secure management authentication mechanisms operates in a way that prevents reuse of authentication data as a means of attack. |
| | FIA_UAU.7 | FIA_UAU.7 stipulates that only success/Fail feedback is given in response to an attempt to establish a secure management session. |
| O.DATA_TRANSFER | FPT_RVM.1 | The objective of O.DATA_TRANSFER is to ensure that the data used in secure management transmissions is secure. FPT_RVM.1 stipulates that all security of the TSF is governed by an SFP and that the SFP must be invoked before access to the TSC is permitted. |
| | FCS_CKM.1(b) | FCS_CKM.1(b) specifies the means by which the key material used to secure the management of the TSF is generated. |
| | FCS_CKM.4 | FCS_CKM.4 states that keys will be destroyed using mechanisms that are compliant with FIPS 140-2. |
| | FIA_UAU.4(a) | FIA_UAU.4(a) stipulates that the single-use secure management authentication mechanisms operates in a way that prevents reuse of authentication data as a means of attack. |
| | FIA_UAU.7 | FIA_UAU.7 stipulates that only success/Fail feedback is given in response to an attempt to establish a secure management session. |
| O.SECURE _RECOVERY | FIA_UAU.4(a) | FIA_UAU.4(a) stipulates that the single-use secure management authentication mechanisms operates in a way that prevents reuse of authentication data as a means of attack. |
| O.AUDIT | FAU_GEN.1 | O.AUDIT requires that specific security relevant events be audited; that audit events are associated with identified users; that the audit log is presented in a comprehensible format and that unauthorised access to the audit log is prohibited.<br><br>FAU_GEN.1 describes the events that are audited by the TSF |
| | FAU_GEN.2<br><br>FAU_STG.1<br><br>FAU_STG.3 | FAU_GEN.2 associates each audit event with an identified user.<br>FAU_STG.1 states that only authorised administrators may modify the audit trail.<br>In order to maintain the audit trail, FAU_STG.3 states that in the event that the audit log becomes full, that new records may overwrite oldest ones |
| | FAU_SAR.1 | FAU_SAR.1 states that there must be a way for users to view the audit trail. |
| | FAU_SAR.3 | FAU_SAR.3 states that the user can organise the audit trail to make it more comprehensible. |
| OE.TIME_SOURCE | FAU_GEN.1 | FAU_GEN.1 requires all audit events to be timestamped. OE.TIME_SOURCE provides such a timestamp. |
| O.PROTECT | FPT_AMT.1 | O.PROTECT requires that the TSF protect itself |

| Security Objectives | TOE Security Functional Requirement | Rationale |
|---|---|---|
| | | against external interference and tampering<br><br>FPT_AMT.1 provides a suite of tests to monitor the correct functioning and integrity of the TSF |
| | FPT_FLS.1 | In the event of failure of the power to the TSF or the connection to the Administration server, FPT_FLS.1 states that the TSF maintains a secure state. |
| | FPT_RCV.1<br><br><br><br>FPT_SEP.1 | FPT_RCV.1 describes a method to return the TSF to an operational state in the event that a user forgets his password or has his account disabled for some other reason.<br>FPT_SEP.1 describes how the TSF is protected by executing in its own separate domain. |
| OE.SECURE_BACKUP | AGD_ADM.1 | AGD_ADM.1 describes how to perform secure backups |
| OE.AVAILABLE_BACKUP | AGD_ADM.1 | AGD_ADM.1 describes how to perform regular backups to enable recovery in the event of TSF failure as a result of attack or some other cause. |
| OE.MANAGED | AGD_ADM.1 | AGD_ADM.1 describes how the TSF should be used in order to fulfil its security objectives, and so provides all of the information required by the personnel responsible for administering the TSF. |
| OE.EASE_OF_USE_ADMIN | AGD_ADM.1 | AGD_ADM.1 describes the administrative procedures required to ensure that the TSF remains secure. |
| OE.AUTH | AGD_USR.1 | AGD_USR.1 describes the credentials that users must keep secure. |
| OE.EASE_OF_USE_USER | AGD_USR.1 | AGD_USR.1 describes the procedures that users must follow in order to maintain the security of the TSF. |

**Figure 17 Justification of the mapping of security objectives to security functional requirements**

## 9.3 TOE Summary Specification Rationale

The Security Objectives do not map directly to the IT Security Functions in a one-to-one fashion, and so a mapping table is included here.

| IT Security Function | Security Objectives |
|---|---|
| TSF.USER_ACCESS_CONTROL | O.ACCESS_CONTROL<br>O.FAULT_TOLERANCE |
| TSF.USER_AUTHENTICATION | O.TRUSTED_PATH<br>O.AUTHORISATION<br>O.CRYPTOGRAPHIC_OPERATIONS<br>O.NO_OBJECT_REUSE |
| TSF.MANAGEMENT_BY_USER | O.EASE_OF_USE_USER |
| TSF.HDD_ENCRYPTION | O.ENCRYPTED_MEDIA<br>O.CRYPTOGRAPHIC_OPERATIONS |
| TSF.HDD_ENC_KEYMAN | O.CRYPTOGRAPHIC_KEYS |
| TSF.ADMIN_ACCESS_CONTROL | O.EFFECTIVE_ADMINISTRATION |
| TSF.SECURE_MANAGEMENT | O.DATA_TRANSFER<br>O.EFFECTIVE_ADMINISTRATION<br>O.CRYPTOGRAPHIC_KEYS<br>O.CRYPTOGRAPHIC_OPERATIONS<br>O.SECURE_RECOVERY<br>O.NO_OBJECT_REUSE<br>OE.ADMINISTRATION_SERVER |

| IT Security Function | Security Objectives |
|---|---|
| TSF.SECURITY_AUDIT | O.AUDIT<br>OE.TIME_SOURCE |
| TSF.PROTECTION | O.PROTECT |

**Figure 18 Mapping of Security Functions to Security Objectives**

Where there is a one-to-many mapping of IT Security Functions to security objectives, the objectives are simply combined together to form the IT Security Functions, that is each the IT Security Functions is the sum of the security objectives that make it up. However, in a number of cases, security objectives contribute to more than one IT Security Function. In these cases, different aspects of the objective contribute to the different IT Security Functions, as follows:

#### 9.3.1.1 O.CRYPTOGRAPHIC_KEYS

TSF.HDD_ENC_KEYMAN: Refers to the symmetric keys used by AES
TSF.SECURE_MANAGEMENT: Refers to the asymmetric keys used by DSS

#### 9.3.1.2 O.CRYPTOGRAPHIC_OPERATIONS

TSF.USER_AUTHENTICATION: The password authentication mechanism
TSF.HDD_ENCRYPTION: Encryption of the hard disk using AES.
TSF.SECURE_MANAGEMENT: Secure management using DSS

#### 9.3.1.3 O.NO_OBJECT_REUSE

TSF.SECURE_MANAGEMENT: Refers to the single-use authentication used to establish a secure management session.
TSF.USER_AUTHENTICATION: Refers to the single-use authentication used during recovery.

The Security Objectives completely satisfy the assumptions, threats and policies, and these objectives in turn are realized by implementing the SFRs. Each objective matches a threat, assumption or policy, or combination of these. There is no security objective that does not address a threat, assumption or policy, or combination of these.

Similarly, each SFR corresponds to one or more security objective and no SFR does not have a matching security objective. Similarly each security objective is matched by one or more SFR.

Sections 9.1 and 9.2 described how the threats, assumptions and policies are addressed by security objectives and how these objectives are met by the SFRs of the TSF. The assurance measures have been shown to match the requirements, and evaluation is required to demonstrate that the measures in fact match the requirements.

### 9.3.2 SOF rationale

There are a number of probabilistic mechanisms used within the TOE, however, specific claims are only made about the password mechanism. Hash functions are used during hard disk key generation and signature verification during secure management session negotiation. However, these are well-defined public algorithms, defined in the FIPS 180 and FIPS 186 standards.

#### 9.3.2.1 Password mechanism

The minimum strength of function claimed for the password mechanism in the TOE is SOF-Medium. This is defined in CC part 1 as follows: *A level of the TOE strength of function where analysis shows that the function provides adequate protection against straightforward or intentional breach of TOE security by attackers possessing a moderate attack potential* .

This claim is based on the strength of the password mechanism. Automated attack is not possible, so the attacker must use a manual approach. SafeBoot recommends a minimum of 5 characters, giving a random chance of success of 1 in 916,132,832. The software is configured to lock up after 10 unsuccessful attempts, this gives a chance of successfully guessing the password at 1 in 91,613,2 83 which is to all intents and purposes impractical and deserving of a rating of SOF-High, but a rating of

SOF-Medium has been chosen as this is believed to be a reasonable strength of function claim for this kind of password mechanism.

Section B.8 of the Common Evaluation Methodology (CEM v2.2) provides a method for calculating the strength of TOE function. Following this procedure, and using guidance from UK CC Interpretation - UK/2.1/005:

If we consider that the principle attack potential is that of a layman manually attacking the password mechanism and we assume that no knowledge of the TOE is needed, and no equipment is used, then the SOF rating is dependent purely on the time taken to defeat the mechanism. If elapsed time and access to the TOE is > 1 month, then this achieves a score of 17 (8 + 9). According to Table B.8.4 this is only SOF-basic (18 is needed for SOF-medium).

However, if an attacker were prepared to spend more than a month trying to subvert the password mechanism, then this individual would not have low motivation (part of the definition of SOF-Basic). Also, an attack of such length would be likely to follow a systematic approach, using for example an automated password generation tool, or a technique to avoid duplicating attempts. Each of these two factors adds 2 to the score, giving a result of 21, and a calculated rating of SOF-medium for the password mechanism.

The rating of SOF-medium can further be justified as any attempt to subvert the password mechanism would have to use the normal logon process, and so any failed attempt would be audited. Also, ten successive unsuccessful attempts would result in the password being disabled of the user account that the attacker was trying to subvert (in CC mode), further adding strength to the mechanism.

## 9.4 PP Claims Rationale

There are no PP claims made in this Security Target.

## 9.5 IT environment rationale

As described in section 7.1.1, the TSF TSF.ADMINISTRATION_SERVER realises the administration server side of the functions, TSF.ADMIN_ACCESS_CONTROL, TSF.SECURE_MANAGEMENT and TSF.SECURITY_AUDIT and are specified as such in sections 5.3.1 and 7. As a result, the rationale for the completeness and correctness of these requirements and functions has already been made in the earlier subsections (9.1, 9.2 and 9.3) of this rationale section.

## 9.6 Security Assurance Requirements Rationale

This ST contains the assurance requirements from the CC EAL4 assurance package and is based on good rigorous commercial development practices. This ST has been developed for a generalized environment.

The TOE is used to protect information assets and it is assumed that possible attackers will have a low attack potential. The Security Objectives for the TOE were derived to resist this class of attacker, and CC EAL4 was found to be sufficient to provide the assurance for the environment.

# 10 Appendix A – Administrative Options

The following options are available from the SafeBoot Administration server to configure the secure attributes of the TOE.

| Object | Category | Options |
|---|---|---|
| Machine | Context menu (right click on machine in machine tree) | 1. Force sync<br>2. Reboot machine<br>3. Lock machine |
| | General | Boot protection is any one of:<br>1. Disable<br>2. Enable<br>3. Remove<br>4. Remove and reboot |

| Object | Category | Options |
|---|---|---|
| | General – Options – each option may be enabled or disabled. | Windows Logon<br>• Require SafeBoot logon (must be set)<br>• Attempt automatic Windows logon<br>• Requires SafeBoot re-logon<br>• Automatically logon as boot user<br>• SafeBoot logon component always active<br>• Set SafeBoot password to Windows password<br>Virus protection<br>• Enable MBR virus protection<br>Miscellaneous<br>• Do not display previous user name at logon<br>• Disable Power Fail Protection during encryption<br>• Allow configuration manager to be closed<br>• Reject suspend/hibernate requests<br>• Disable checking for Autoboot |
| | Encryption | Only full encryption is supported in CC mode.<br><br>Recovery key size (64, 128, 192 or 256 bits) |
| | Synch - the following options may be enabled or disabled: | • Automatically resynchronise every *nn* minutes<br>• Allow local resynchronisation<br>• Resynchronise when RAS connection is detected<br>• Synchronise time with database<br>• Disable synchronisation of files<br>• Allow remote controlled resynchronisation, address *nnn.nnn.nnn.nnn* , port *nnnn*<br>• Disable access if not synchronised for *nn* days<br>• Delay synch at boot for *nn* minutes plus random up to *nn* minutes |
| | Screen saver | Options<br>The following options may be enabled or disabled:<br>• Allow user access to the Windows s creen saver options<br>• Run screen saver if token is removed (if supported)<br>• Set SafeBoot screen saver as default<br>• Allow logon of administrators greater than level *n*<br>• Set screen saver inactivity timeout (minutes) *n* |
| User | Context menu (right click on user in the u ser tree) | The following options are available:<br>• Create User<br>• Rename<br>• Delete<br>• Create Token<br>• Reset Token<br>• Set SSO details<br>• View Audit<br>• Reset to Group Configuration<br>• Create copy<br>• Properties (brings up all of the other user options – below) |
| | General | User accounts are either enabled or disabled and can be enabled indefinitely or for a fixed calendar period (from a start date to an end date). |

| Object | Category | Options |
|---|---|---|
| | Passwords | The following options may be enabled:<br>Password change<br>• Force change if '12345'<br>• Prevent change<br>• Enable password history *nn*<br>• Require change after *nn* days warn *nn* days before<br>Incorrect passwords<br>• Timeout password entry after three invalid attempts Maximum disable time *nn* minutes<br>• Invalidate password after *nn* attempts |
| | Password templates | SafeBoot enforces minimum and maximum password lengths:<br>• Minimum length *nn*<br>• Maximum length *nn*<br><br>Password content enforcement may be enabled (each type of control is optional, and all may be employed if required):<br>• Password must have a minimum of *n* alpha characters<br>• Password must have a minimum of *n* numeric characters<br>• Password must have a minimum of *n* alphanumeric characters<br>• Password must have a minimum of *n* symbol characters<br><br>Password content restrictions may be enabled (each type of control is optional, and all may be employed if required):<br>• No anagram s<br>• No sequences<br>• No palindromes<br>• No simple words<br>• Can't be user name |