Check Point Endpoint Security
Full Disk Encryption

Security Target

ST Version 2.4

June 22, 2009

Prepared for:

5 Ha'Solelim St.

Tel Aviv, Israel 67897

Prepared by:

Metatron Ltd.

66 Yosef St.,

Modiin, Israel 71724

Table of Contents

1. Security Target Introduction

This section presents the following information:

- Identifies the Security Target (ST) and Target of Evaluation (TOE);
- Specifies the ST conventions and ST conformance claims; and
- Describes the ST organization.

1.1 Security Target, TOE and CC Identification

ST Title – Check Point Endpoint Security Full Disk Encryption Security Target

ST Version – Version 2.4

ST Date - 6/22/2009

TOE Identification – Pointsec PC 6.3.1

Evaluation Assurance Level (EAL) – EAL4 augmented with ALC_FLR.1

Common Criteria Identification – Common Criteria for Information Technology Security Evaluation, Version 2.3, August 2005. International Standard – ISO/IEC 15408:2005.

Keywords – disk encryption, pre-boot authentication, security target, EAL 4 augmented, Pointsec.

1.2 CC Conformance Claims

This 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) augmented with ALC_FLR.1-Basic Flaw Remediation

1.3 TOE Summary

The TOE is a disk encryption software based product that can be centrally administered throughout the enterprise. The TOE employs both pre-boot authentication and transparent disk encryption to provide protection of information resources stored on fixed media in a workstation or a laptop.

1.4 Strength of Environment

The TOE, Pointsec PC 6.3.1, has been developed for an operating environment with a moderate level of risk to identified assets. The assurance requirements of EAL 4 augmented with ALC_FLR.1 and the minimum strength of function of *SOF-medium* were chosen to be consistent with that level of risk.

1.5 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the Security Target.

1.5.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter in parenthesis placed at the end of the component. For example FDP_ACC.1 (a) and FDP_ACC.1 (b) indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold italics and are surrounded by brackets (e.g., [***assignment***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using italics and are surrounded by brackets (e.g., [*selection*]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions and the application of interpretations.

1.5.2 Terminology

The following terminology is used in the Security Target:

- **Administrator:** Accounts at this level have limited authority in the administration of the TOE (according to what has been defined in the system settings). The Administrator can typically view logs and provide remote help. Administrators can not raise their own authorization level.
- **Authentication data:** Information used to verify the claimed identity of a user.
- **Authorized administrators:** A term used to encompass both the Administrator and System Administrator roles defined in this ST.
- **Authorized users:** A term used to describe all users that interact with the TOE that have a unique identifier. This includes the non-privileged set of users and all others within the Administrator and System Administrator groups.
- **Disk Partition:** A logical division of a hard disk. Each partition can be formatted for a different file system. A partition must be completely contained on one physical disk. The Master Boot

Record for a physical disk can contain up to four entries for partitions, including one extended partition, which can be further subdivided into logical volumes, allowing for more than four partitions on one physical disk.

- **File Share:** A storage resource where installation files, profiles, recovery files and software updates can be stored. System Administrators are able to utilize the share to install and configure the system, delegate authorization, modify the system for local conditions, and assign the properties and authorization of individual users by using profiles.

- **FIPS 140-2:** Federal Information Processing Standards Publication published by the National Institute of Standards and Technology (NIST) to define security requirements for cryptographic modules.

- **Fixed password authentication:** A normal password authentication mechanism. The administrator can make changes to the default requirements for passwords.

- **Group Authority Level:** a numeric authorization (0-9) associated with user groups and with system settings, defining a restriction for the objects that each user group may administer.

- **Identity:** A representation uniquely identifying an authorized user.

- **One-time Login authentication:** An authentication mechanism whereby a user who normally authenticates with a smart card is granted temporary, one-time access to the TOE. See Remote Help authentication mechanisms.

- **Partition key (K_P):** A symmetric encryption key that is used by the TOE to encrypt individual partitions on a hard drive.

- **Remote Help authentication mechanism:** A secondary authentication mechanism, only used in special circumstances, where the user requests login assistance from authorized personnel over the phone. This mechanism uses a challenge-response sequence that is read over the phone to provide the user authorization for access to the TOE. There are two types of Remote Help, One-time login and Remote Password Change. These mechanisms provide temporary authentication to the TOE when normal authentication is not possible.

- **Remote Password Change authentication:** This type of authentication allows a user to change a forgotten password during the login process with the help of authorized personnel over the phone. This is also the basis for remotely unlocking a locked user account.

- **Smart card authentication:** Authentication mechanism employed by the TOE that utilizes smart cards to store credentials for the user that can only be accessed with a PIN, known only to the owner of the card.

- **System Administrator:** The highest authorization level in the administration of the TOE. This role can: create and administer profiles, configure system settings, add and remove administrators and users, configure settings for administrators and users, and provide remote assistance to users who are locked out or have forgotten their passwords.

- **System Area:** A protected area on each partition where TOE-specific security information is stored. The System Area is hidden from the OS and is under TOE control.

- **User Key (K_u):** Symmetric encryption key that is used to decrypt the Partition Key.

- **Users:** Any external user that interacts with the TOE.

1.5.3 Acronyms

The acronyms used within this Security Target:

AES	Advanced Encryption Standard
BIOS	Basic Input/Output System
BS	Boot Sector
CBC	Cipher Block Chaining
DES	Data Encryption Standard
EAL	Evaluation Assurance Level
ECB	Electronic Codebook
EW	Enterprise Workplace
FIPS	Federal Information Processing Standards Publication
GAL	Group Authority Level
GB	Gigabyte
I/O	Input/Output
MAC	Message Authentication Code
MBR	Master Boot Record
NIST	National Institute of Standards and Technology
PCMC	Pointsec for PC Management Console
PBE	Pre-Boot Environment
PP	Protection Profile
PRNG	Pseudo Random Number Generator
RSA	Rivest Shamir Adleman (public key algorithm)
SF	Security Functions
SFR	Security Functional Requirements
ST	Security Target
TOE	Target of Evaluation
TSF	Target of Evaluation Security Functions
TSP	TOE Security Policy

1.5.4 Security Target Overview and Organization

The Security Target contains the following additional sections:

- TOE Description (Section 2): Provides an overview of the TOE security functions and boundary.
- Security Environment (Section 3): Describes the threats, organizational security policies and assumptions that pertain to the TOE.
- Security Objectives (Section 4): Identifies the security objectives that are satisfied by the TOE and the TOE environment.
- IT Security Requirements (Section 5): Presents the security functional and assurance requirements met by the TOE.
- TOE Summary Specification (Section 6): Describes the security functions provided by the TOE to satisfy the security functional requirements and objectives.
- Protection Profile Claims (Section 7): Presents the rationale concerning compliance of the ST with any Protection Profiles
- Rationale (Section 8): Presents the rationale for the security objectives, requirements, and TOE summary specifications as to their consistency, completeness and suitability.

2. TOE Description

2.1 Product Overview

Pointsec PC 6.3.1 is a disk encryption product that can be centrally administered throughout the enterprise. The TOE employs both boot authentication and transparent disk encryption to provide protection of information resources stored on fixed media in a workstation or a laptop. Pointsec PC is a software based security product, for the Windows based PC platform. The product contains an embedded cryptographic module that is certified against FIPS 140-2 Level 1 (certificate#770), used for all cryptographic functions.

The TOE prevents unauthorized access to the machine itself, and provides further security by encrypting everything on the machine. This is accomplished through user authentication linked to boot protection, which in turn enables information to be automatically encrypted and decrypted. Strong user authentication and boot protection are necessary components to this system.

Full hard drive encryption offers several key advantages relative to file encryption. The most important is that full hard drive encryption is automatic and transparent to the user. Not only does this decrease user involvement and training requirements, but it also creates the foundation for enforceable security. In addition, full hard drive encrypts the system and temp files that often contain sensitive data but are missed by file encryption. Even removing the drive itself does not give access to any file or directory structure. Finally, hard drive encryption is performed sector by sector without creating temp or backup files; as a result, large files will decrypt without delay whereas file encryption is normally much slower. Full hard drive encryption also avoids such time consuming tasks as secure deletes of temp files or work files in clear text, and obviates the need to do a full delete on disks to be discarded.

The TOE encrypts the entire disk sector by sector including the system files, temp files, deleted files and unused space. The encryption is user transparent and automatic, so there is no need for user intervention or user training. Because the encryption occurs in the background without noticeable performance lost, there is no user downtime.

Figure 1 below illustrates the combination of full-disk encryption and boot protection implemented by Pointsec PC 6.3.1, as compared to an unprotected system on a system employing file encryption.

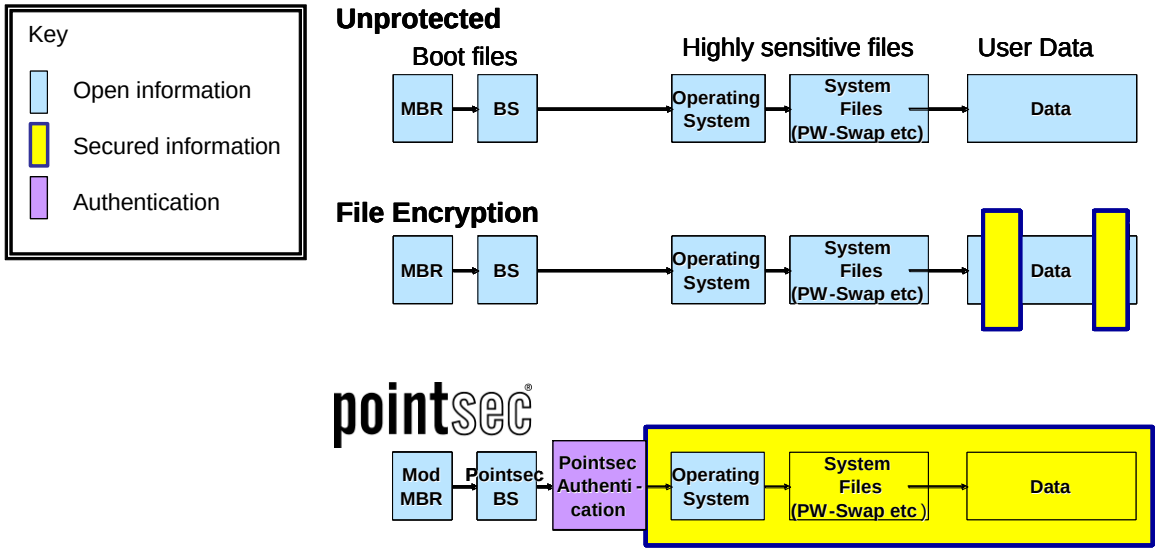

Figure 1: Pointsec PC Full-Disk Encryption

2.2 Summary of TOE Security Functionality

2.2.1 Identification and Authentication

The TOE supports multiple user authentication mechanisms enabling the administrator to assign appropriate authentication requirements for the intended environment, including:

- fixed password (username/password),
- smart card (and USB token with embedded smart card) based authentication (smart card/PIN).
- Remote Help authentication (username/phone identification/TOE challenge/Admin response). Remote Help is divided into two types, one-time login and remote password change. These provide a way to authorize a user to login when the normal authentication process can not be performed, such as when the user forgets their smart card at home, or a fixed password has been forgotten.

Where a smart card is used for authentication, the card and reader (or token) are part of the IT environment.

User authentication is done in the pre-boot environment and the operating system will not boot up unless an authorized user is authenticated. In addition, administrators authenticate using the same mechanisms as above prior to gaining access to the Pointsec for PC Management Console application (see below).

2.2.2 Security Management

Pointsec for PC administration is designed to enable central control of policy and security settings, decentralized deployment and day-to-day administration. Pointsec for PC should be administered using several different levels of authority. It can be administered from the Pointsec for PC Management Console (PCMC) on any computer that has the product installed on it. This gives the administrators control and easy access to higher-level functionality without being tied to one computer.

Security management includes managing the following items: authentication data, group memberships, audit data, and cryptographic functions. The configuration information is stored in encrypted profiles that then could be placed on a file share. Profiles always contain system information. System information determines general TOE behavior, e.g. paths to central log file, recovery file, and update profile directories. In addition, profiles may contain group and user account information, including user and group permissions and user authentication settings. An installation profile is used for initial installation of the TOE on a user's workstation. Installation profiles also determine the cryptographic protection (e.g. encryption algorithms) that is applied on the workstation. After initial installation, administrators can create and deploy update profiles that contain changes to selected system, user, and group settings, as depicted in Figure 2 below.

Another function of central administration is Remote Help, which provides secure access if users lock themselves out. Available for administrators via the PCMC is also the Recovery Utility which provides secure recovery of encrypted information in case of for example an operating system crash.

2.2.3 Self Protection

Pointsec for PC implements a specific set of security mechanisms to ensure that security functions cannot be bypassed or tampered with.

To prevent bypassing of the TOE security functions, the TOE takes control of the Boot Sector of the boot partition, which prevents access to the system without successful authentication. The Boot-code is checked

for the presence of debugging tools at each step of the loading process. If suspicious code is detected, the boot process will stop. Within the Windows operating system, Pointsec for PC functions as a kernel mode process, restricting access to its execution space and memory. When the TOE starts (from Power on) it has its own OS and is later handing over control to Windows after it has authenticated the user and recreated the encryption key. During the time Pointsec is in control Windows security does not matter.

To ensure correct operations of the cryptographic operations the TOE runs cryptographic self-test on the FIPS 140-2 algorithms.

2.2.4 Auditing

The TOE collects audit data and provides an interface for authorized administrators to review audit logs. Audit information generated by the system includes date and time of the event, user ID that caused the event to be generated, computer where the event occurred, and other event specific data. The TOE also restricts log access to authorized users.

2.2.5 Cryptographic functionality

Cryptographic keys are generated, accessed, protected, and destroyed in accordance with requirements defined by FIPS 140-2 Level 1. The TOE supports cryptographic operations such as data and key encryption/decryption and cryptographic self-tests. For user authentication using smart cards the TOE uses RSA public key cryptography.

2.2.6 Fault Tolerance

When a Pointsec PC workstation/laptop loses contact with the file share server, the TOE provides the administrator with the capability to identify additional three servers for redundancy. As a result, if a server is offline, or the workstation/laptop is unable to contact it, the workstation/laptop will attempt to communicate with one of the other identified servers. Even if no storage resource is accessible the TOE will continue to operate as normal.

2.2.7 Trusted Path

For initial logon, a user must invoke a trusted path in order to ensure the protection of identification and authentication information. The trusted path is invoked by a system reset which is always captured by the TOE (i.e. it cannot be intercepted by an untrusted process).

2.3 TOE Boundary

2.3.1 Logical Boundaries

The Logical Boundaries of the TOE is defined by the interfaces necessary to implement the TOE security functions. The TOE interfaces are:

- The security management PCMC interface available for administrators to enable central control of policy and security settings, decentralized deployment and day-to-day administration.
- The authentication interfaces (both pre-boot and in PCMC) available for users and administrators to provide logon data for authentication.

- The disk interface for performing cryptographic operations as data is read and written to the storage media.

- The network interface for accessing the file share where recovery files and log files are written, and from where installation and update profiles and software updates are distributed.

2.3.2 Physical Boundaries

Since the TOE is a software product, its physical boundary is defined by the physical case of the computer where it is installed. The TOE can be installed on any x86 compatible computer running Microsoft Windows 2000, Microsoft Windows XP Professional and Windows XP Tablet PC Edition, Microsoft Windows Server 2003 and Microsoft Windows Vista. Microsoft .NET Framework 2.0 or later is required for PCMC.

Installation files, recovery files, update profiles, and software updates can be stored on a storage resource outside of the TOE's physical boundary (e.g. file server), as shown in figure 2. This provides member workstations/laptops with a central point for storage. All security related files (profiles, central log files, and recovery files) are encrypted before they are stored on the server. Access to the server itself is configured through the server (instructions are detailed in the installation guide and administrator's guide). The server where the file share resides is not part of the TOE but the IT-environment. No components of the TOE need to be installed on the file server. Separate instances of the TOE are installed on the administrator's workstation and each of the laptops or PCs included in the system.

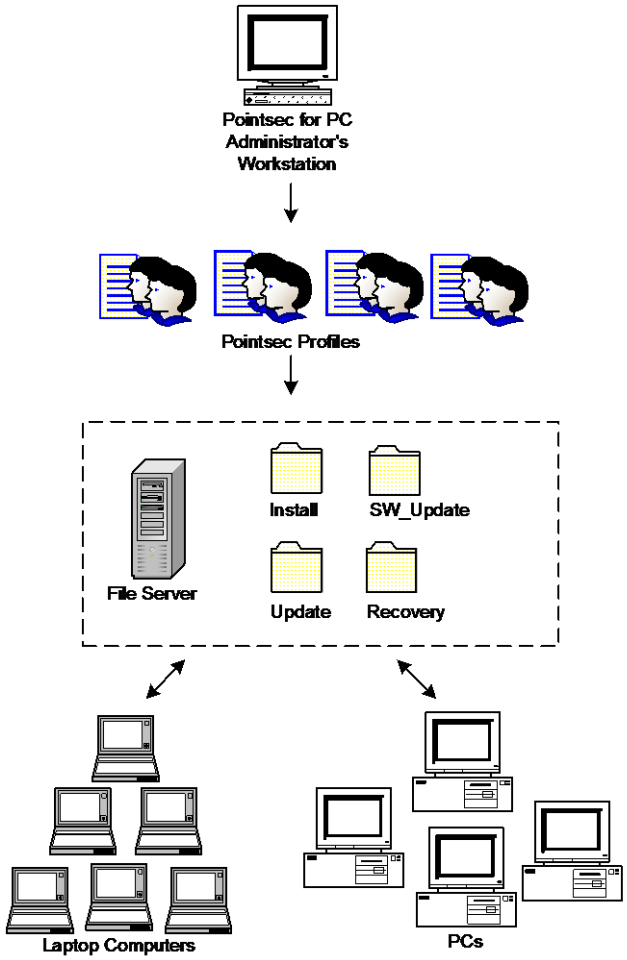

Figure 2: TOE's Physical Environment

2.3.3 Functionality Excluded from the TOE Evaluated Configuration

TOE administration guidance lists the following Pointsec PC 6.3.1 product functionality that is unavailable in the Common Criteria evaluated configuration:

- Wake on LAN – a feature whereby the computer can be automatically started when it receives a specific signal from the network. The administrator can then perform maintenance on the computer without having to visit its physical location.
- Windows Integrated Logon - enables a user to bypass pre-boot authentication at startup.
- Password synchronization – provides automatic synchronization between Windows and Pointsec PC 6.3.1 pre-boot passwords.
- Dynamic token authentication – the Pointsec PC 6.3.1 product supports dynamic one-time password generation tokens, based on a challenge/response protocol. Dynamic tokens are not supported in the evaluated configuration. Note that USB tokens (with embedded smart card) are supported in the evaluated configuration.

3. Security Environment

The TOE security environment describes the security aspects of the environment in which the TOE is intended to be used and the manner in which it is expected to be employed.

The statement of TOE security environment defines the following:

- Threats that the product is designed to counter
- Assumptions made on the operational environment and the method of use intended for the product,
- Organizational security policies with which the product is designed to comply.

3.1 Threats to Security

Threats are undesirable events and are characterized in terms of a threat agent, a presumed attack method, vulnerabilities that are the foundation for the attack, and identification of the asset under attack.

Threat agents can be categorized as either individuals who have not been granted the right to access the system (unauthorized users) or authorized users of the TOE that have been granted the right to access the system, but may attempt to access assets protected by the system to which they do not have permission to access.

Assets comprise the information stored, processed or transmitted by the TOE. The term “information” is used here to refer to all data held within a system, including data in transit between separate parts of the TOE.

In general, the ***threat agents*** are assumed to have an attack potential of ***medium***. As a result, the TOE has been evaluated with the assumption that a potential attacker would have a medium level of expertise, access to a medium level of resources, and also have a medium level of motivation.

Following are the threats countered by the TOE:

T.REMOVE_DISK	An unauthorized user with physical access to the system may remove a systems hard drive to subvert authentication mechanisms allowing them to gain unauthorized access to information contained on the hard drive.
T.SUBVERT	An unauthorized user with physical access to the system may subvert the system's normal boot process allowing them to access information assets contained on the system.
T.ACCESS	An authorized user of the TOE may access information without having permission from the person who owns, or is responsible for, the information.
T.TRANSIT	An unauthorized user may eavesdrop on communications between the TOE and its environment allowing them to gain unauthorized access to information.
T.TSF_DATA	Internal configuration data or other trusted data (such as registry settings) may be tampered with by unauthorized users.
T.AUDIT_CORRUPT	Unauthorized users may tamper with audit data by gaining unauthorized access to the audit trail.
T.RECORD_ACTIONS	An unauthorized user may perform unauthorized actions that go undetected.
T.SYSACC	An unauthorized user may gain unauthorized access to the system and act as the administrator or other authorized users.
T.SPOOF	A hostile entity masquerading as the IT system may receive unauthorized access to authentication data from authorized users who incorrectly believe they are communicating with the IT system during attempts by a user to initially logon.
T.UNAUTH_MOD	An unauthorized user may cause the modification of the security enforcing functions in the system (executable code), and thereby gain unauthorized access to system and user resources.

3.2 Organization Security Policies

Following are the Organizational Security Policies enforced by the TOE:

P.ACCOUNTABILITY	Users of the system shall be held accountable for their security relevant actions within the system.
P.MANAGE	The system must provide authorized administrators with utilities to effectively manage the security functions of the TOE.
P.CRYPTO_KEYS	Cryptographic keys will be generated, accessed, protected, and destroyed in accordance with requirements defined by FIPS 140-2 Level 1
P.CRYPTO_OPS	All cryptographic operation performed by the system will be compliant with the requirements of FIPS 140-2 (product evaluation), FIPS 46-3 (3DES) and FIPS 197 (AES).

P.AUTH_USERS	Only those users who have been authorized access to information within the system may access the system.
P.TRANSIT	The system must have the ability to protect system data in transmission between distributed parts of the protected system
P.FAULT_TOLERANCE	The system must ensure that security functions continue to operate if contact with the file share is lost.

3.3 Secure Usage Assumptions

This section describes the aspects of the operating environment in which the TOE is intended to be used—including personnel and physical assumptions of the environment. The TOE is assured of providing effective security measures in its intended environment only if it has been delivered, installed, and administered as intended.

3.3.1 Personnel Assumptions

A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NO_EVIL	The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.
A.TRAINED_STAFF	Authorized TOE users and administrators are trusted to follow the guidance provided for the secure operation of the TOE.
A.AUTH_DATA	Authorized users of the TOE will keep all their authentication data private.

3.3.2 IT Environment Assumptions

A.SERVER	The TOE's IT environment will provide a storage resource for the management of the TOE installation files, recovery files, update profiles, and software updates.
A.TIME	The TOE's IT environment will provide a reliable time source to enable the TOE to timestamp audit records.

3.3.3 Non-IT Environment Assumptions

A.PHONE_DATA	The system personnel maintain a TOE-independent database containing a list of authorized TOE users and administrators along with unique, non-TOE authentication data that can be used to verify identity over a phone connection (i.e. no video, only voice communications) for the purposes of providing Remote Help authentication to authorized TOE users.

4. Security Objectives

This section defines the security objectives of the TOE and its supporting environment. Security objectives, categorized as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. All of the identified threats and organizational policies are addressed under one of the categories below.

4.1 Security Objectives for the TOE

O.AUTHORIZATION	The TSF must ensure that only authorized 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.MEDIA_ACCESS	The TSF must provide complete hard drive encryption to protect information assets from unauthorized users that have gained physical access to the TOE's storage media.
O.MANAGE	The TOE must allow administrators to effectively manage the TOE and its security functions, and must ensure that only authorized 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 a unique user. The TSF must present this information in a readable format to authorized users and ensure that only authorized users are able to access this information.
O.PROTECT	The TSF must protect its own data and resources and must maintain a domain for its own execution that protects it from external interference or tampering.
O.TRUSTED_PATH	The TSF must provide the capability to allow users to ensure they are communicating with the TSF 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 distributed parts of the system.
O.CRYPTO_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 (product evaluation).
O.CRYPTO_OPS	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 (product evaluation), FIPS 46-3 (3DES) and FIPS 197 (AES).
O.FAULT_TOLERANCE	The TSF must continue to enforce security policies if contact with the file share is lost.

4.2 Security Objectives for the Environment

4.2.1 Security Objectives for the IT Environment

OE.TIME_SOURCE	The TOE's IT environment must provide a reliable time source for the TOE to provide accurate timestamps for audit records.
OE.SERVER	The TOE's IT environment must provide a file share server to be used as a distribution point for recovery, update and installation files.
OE.SMART_CARD	The TOE's IT environment must be able to provide a secure cryptographic token in support of user authentication.

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. Also, an independent database of authentication data is maintained for phone-based authorization.
OE.AUTH	Those responsible for the TOE must ensure that all access credentials, such as passwords or smart cards, are protected by users in a manner that maintains IT security objectives.

5. IT Security Requirements

5.1 TOE Security Functional Requirements

This section specifies the security functional requirements (SFRs) for the TOE. This section organizes the SFRs by CC class. Table 1 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 1: TOE SFRs and associated operations

Functional Class	Functional Components	ST Operation
Security Audit (FAU)	FAU_GEN.1 - Audit data generation	Selection, Assignment
	FAU_GEN.2 - User identity association	None
	FAU_SAR.1 - Audit review	Assignment
	FAU_SAR.2 - Restricted audit review	None
	FAU_SAR.3 - Selectable audit review	Selection, Assignment
	FAU_STG.1 - Protected audit trail storage	Selection
Cryptographic Support (FCS)	FCS_CKM.1 - Cryptographic key generation	Assignment
	FCS_CKM.4 - Cryptographic key destruction	Assignment
	FCS_COP.1(a) - Cryptographic operation (data encryption and decryption)	Iteration, Assignment
	FCS_COP.1(b) - Cryptographic operation (cryptographic key encryption and decryption)	Iteration, Assignment
	FCS_COP.1(c) - Cryptographic operation (asymmetric cryptographic operations)	Assignment
	FCS_COP.1(d) - Cryptographic operation (one-way encryption)	Assignment
	FCS_COP.1(e) - Cryptographic operation (integrity of audit data)	Assignment
Identification and Authentication (FIA)	FIA_AFL.1 - Authentication failure handling	Assignment
	FIA_ATD.1 – User attribute definition	Assignment
	FIA_SOS.1 - Verification of secrets	Assignment

Functional Class	Functional Components	ST Operation
	FIA_UAU.2 - User authentication before any action	None
	FIA_UAU.4 - Single-use authentication mechanisms	Assignment
	FIA_UAU.5 - Multiple authentication mechanisms	Assignment
	FIA_UAU.7 - Protected authentication feedback	Assignment
	FIA_UID.2 – User identification before any action	None
Security Management (FMT)	FMT_MOF.1 - Management of security functions behavior (Identification and authentication)	Selection, Assignment
	FMT_MTD.1(a) - Management of TSF Data (security-relevant roles)	Selection, Assignment, Iteration
	FMT_MTD.1(b) - Management of TSF Data (authentication data)	Selection, Assignment, Iteration
	FMT_MTD.1(c)- Management of TSF Data (password policy)	Selection, Assignment, Iteration
	FMT_MTD.1(d)- Management of TSF Data (authentication failure)	Selection, Assignment, Iteration
	FMT_MTD.2 - Management of limits on TSF Data (authentication failure)	Assignment
	FMT_REV.1 – Revocation	Selection, Assignment
	FMT_SAE.1 - Time-limited authorization	Assignment
	FMT_SMF.1 – Specification of Management Functions	Assignment
	FMT_SMR.1 - Security roles	Assignment
Protection of the TSF (FPT)	FPT_FLS.1 – Failure with preservation of secure state	Assignment
	FPT_ITT.1 – Basic internal TSF data transfer protection	Selection

Functional Class	Functional Components	ST Operation
	FPT_RVM.1 - Non-bypassability of the TSP	None
	FPT_SEP.1 - TSF domain separation	None
	FPT_TST.1 - Self testing	Selection
Resource Utilization (FRU)	FRU_FLT.1 - Degraded fault tolerance	Assignment
TOE Access (FTA)	FTA_SSL.1 – Session locking	Assignment
Trusted Path / Channels (FTP)	FTP_TRP.1 - Trusted Path	Selection, Assignment

5.1.1 Audit (FAU) Requirements

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;

c) and [

- ***All attempts to use authentication mechanisms***
- ***All attempts to access locked user accounts***
- ***Account locked out due to exceeding the maximum number of unsuccessful logon attempts***
- ***All changes to the user database***
- ***Use of the Remote Help feature***]

Application Note: in the context of this ST, start-up and shutdown of the audit functions are synonymous with start-up and shutdown of the TOE, as the audit functions start-up automatically, and the TOE does not provide a management function for shutting them down. Shutdown of the TOE is not controlled by the TSF and therefore not considered a required auditable event.

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 information***].

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_SAR.1 - Audit review

FAU_SAR.1.1 The TSF shall provide [***authorized administrators***] with the capability to read [***audit information***] from the audit records.

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

FAU_SAR.2 – Restricted audit review

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read access.

FAU_SAR.3 - Selectable audit review

FAU_SAR.3.1 The TSF shall provide the ability to perform [*searches and sorting*] of audit data based on [***user identity (sorting), audit time stamp (search and sort)***].

FAU_STG.1 – Protected audit trail storage

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

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

5.1.2 Cryptographic Support (FCS)

FCS_CKM.1 - Cryptographic key generation

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [***FIPS 140-2 PRNG***]

and specified cryptographic key sizes [

a) ***168 bits (Triple DES)***

b) ***256 bits (AES)***]

that meet the following: [***FIPS 46-3 (Triple DES) and FIPS 197 (AES)***].

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 [***zeroization***] that meets the following: [***FIPS 140-2 Level 1, Section 4.7.6 Key Zeroization***].

FCS_COP.1(a) - Cryptographic operation (data encryption and decryption)

FCS_COP.1.1(a) The TSF shall perform [***data encryption and decryption***] in accordance with a specified cryptographic algorithm [***listed below***] and key sizes [***listed below***] that meet the following: [***listed below***].

a) Triple DES

- ***Key size: 168 bits***
- ***Based on standard: FIPS 46-3***
- ***Modes of operation: CBC***

b) AES

- ***Key size: 256 bits***
- ***Based on standard: FIPS 197***
- ***Modes of operation: CBC***

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

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

Application note: Overlapping encryption is done in order to achieve diffusion and is described in detail in 6.1.1.

FCS_COP.1(c) - Cryptographic operation (asymmetric cryptographic operations)

FCS_COP.1.1(c) The TSF shall perform [***asymmetric key encryption for smart card authentication***] in accordance with a specified cryptographic algorithm [***RSA***] and key sizes [***1024, 2048 and 4096 bits***] that meet the following: [***RSA PKCS#1 version 1.5***].

FCS_COP.1(d) - Cryptographic operation (one-way encryption)

FCS_COP.1.1(d) The TSF shall perform [***one-way encryption of passwords***] in accordance with a specified cryptographic algorithm [***AES in ECB mode, using the password itself as key***] and key size [***256 bits***] that meet the following: [***FIPS 197***].

Application note: Since the password is one-way encrypted using itself as key (i.e. no static key is involved) then it is required to use passwords of at least 8 characters (64 bits), to reach strength-of-function medium. The maximum length of passwords in the TOE is 32 characters (256 bits). The entropy of the key is dependent on the length of the password, passwords shorter than 32 characters are padded so that they can be used in AES 256 bit mode.

FCS_COP.1(e) - Cryptographic operation (integrity of audit data)

FCS_COP.1.1(e) The TSF shall perform [***integrity protection of audit data***] in accordance with a specified cryptographic algorithm [***HMAC SHA-256***] and key size [***256 bits***] that meet the following: [***FIPS 198***].

5.1.3 Identification and Authentication (FIA)

FIA_AFL.1 – Authentication failure handling

FIA_AFL.1.1 The TSF shall detect when [***an authorized administrator specified number of***] unsuccessful authentication attempts occur related to [***fixed password 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 for either a specified duration or until unlocked by an authorized administrator (as specified by an authorized administrator)***].

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 identifier;***

b) ***Security management roles and permissions; and***

c) ***Authentication data***].

FIA_SOS.1 - Verification of secrets

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet [***the following:***

a) ***For each attempt to use the fixed password authentication mechanism, the probability that a random attempt will succeed is less than one in 100,000,000,000.***

b) ***For multiple attempts to use the fixed password authentication mechanism during a one-minute period, the probability that a random attempt during that minute will succeed is less than one in 10,000,000,000***].

Application note: The TOE also supports Smart Card and Remote Help authentication mechanisms. These mechanisms are not dependent on a user-selected secret and are therefore not applicable to FIA_SOS.1. The selection of the Smart Card PIN is controlled by the IT environment, in accordance with OE.AUTH.

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 the behalf of that user.

FIA_UAU.4 - Single-use authentication mechanisms

FIA_UAU.4.1 The TSF shall prevent reuse of authentication data related to [***Remote Help Authentication Mechanism***].

FIA_UAU.5 - Multiple authentication mechanisms

FIA_UAU.5.1 The TSF shall provide [

a) ***Fixed Password Mechanism***

b) ***Smart Card Authentication Mechanism***

c) ***Remote Help Authentication Mechanism***]

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the [***authentication mechanism specified by the authorized administrator***].

FIA_UAU.7 - Protected authentication feedback

FIA_UAU.7.1 The TSF shall provide only [***obscured feedback***] 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 the behalf of that user.

5.1.4 Security Management (FMT)

FMT_MOF.1- Management of security functions behavior (Identification and authentication)

FMT_MOF.1.1 The TSF shall restrict the ability to [*modify the behavior of*] the function [***identification and authentication***] to [***the System Administrator Role***].

FMT_MTD.1(a) - Management of TSF Data (security-relevant roles)

FMT_MTD.1.1(a) The TSF shall restrict the ability to [*modify*] the [***security management roles and permissions for users***] to [***the System Administrator Role***].

FMT_MTD.1(b) - Management of TSF Data (authentication data)

FMT_MTD.1.1(b) The TSF shall restrict the ability to [*modify*] the [***authentication data***] to [***authorized administrators, and users (for their own authentication data)***].

FMT_MTD.1(c) - Management of TSF Data (password policy)

FMT_MTD.1.1(c) The TSF shall restrict the ability to [*modify*] the [***requirements for password composition and length, or specification of other authentication mechanisms (such as smart card authentication)***] to [***the System Administrator Role***].

FMT_MTD.1(d) - Management of TSF Data (authentication failure)

FMT_MTD.1.1(d) The TSF shall restrict the ability to [*modify*] the [***settings for handling authentication failures***] to [***the System Administrator Role***].

FMT_MTD.2 - Management of limits on TSF Data (authentication failure)

FMT_MTD.2.1 The TSF shall restrict the specification of limits for [***the unsuccessful authentication attempts threshold***] to [***the System Administrator Role***].

FMT_MTD.2.2 The TSF shall take the following action, if the TSF data are at, or exceed the indicated limits: [***disable the user account for either a specified duration or until unlocked by an authorized administrator (as specified by the System administrator)***].

FMT_REV.1 – Revocation

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the [*users*] within the TSC to [***the System Administrator Role***].

FMT_REV.1.2 The TSF shall enforce the rules: [***Revocation will take place on the next login of the user***].

FMT_SAE.1 - Time-limited authorization

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for [***fixed password authentication data***] to [***the System Administrator Role***].

FMT_SAE.1.2 For each of these security attributes, the TSF shall be able to [***deny access to the associated user account***] after the expiration time for the 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: [

a) ***management of user accounts (create, delete, modify)***

b) ***Remote Help assistance***

c) ***Management of security settings***

d) ***Review of audit trail***].

FMT_SMR.1 - Security roles

FMT_SMR.1.1 The TSF shall maintain the roles: [

a) ***System Administrator***

b) ***Administrator; and***

c) ***User***].

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

Application note: The roles defined in FMT_SMR.1 correspond to recommendations given in the TOE Administrator's Guide for grouping users in a three-level hierarchy for administration. These roles are then used in this ST for expressing security administration restriction requirements. The TOE provides a granular system for assigning permissions to users that can be configured to support this hierarchy or alternative administrator permission sets. See section 6.1.2 below for further details.

5.1.5 Protection of the TSF (FPT)

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: [***contact is lost with the file share***].

FPT_ITT.1 – Basic internal TSF data transfer protection

FPT_ITT.1.1 The TSF shall protect TSF data from [*disclosure, modification*] when it is transmitted between separate parts of the TOE.

FPT_RVM.1 - Non-bypassability of the TSP

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

FPT_SEP.1 – TSF domain separation

FPT_SEP.1.1 The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

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

FPT_TST.1 - TSF testing

FPT_TST.1.1 The TSF shall run a suite of self tests [*during initial start-up*] to demonstrate the correct operation of the TSF.

FPT_TST.1.2 The TSF shall provide authorized users with the capability to verify the integrity of the TSF data.

FPT_TST.1.3 The TSF shall provide the authorized users with the capability to verify the integrity of stored TSF executable code.

5.1.6 Resource Utilization (FRU)

FRU_FLT.1 - Degraded fault tolerance

FRU_FLT.1.1 The TSF shall ensure the operation of [***normal user functionality***] when the following failures occur: [***access to file share is lost***].

5.1.7 TOE Access (FTA)

FTA_SSL.1 – Session locking

FTA_SSL.1.1 The TSF shall lock an interactive session after [***15 minutes***] 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.1.2 The TSF shall require the following events to occur prior to unlocking the session: [***re-authentication***].

Application note: the term 'interactive session' in FTA_SSL.1.1 refers to the use of the PCMC interface.

5.1.8 Trusted Path / Channels (FTP)

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 the 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, and augmented by ALC_FLR.1. No operations have been applied to the TOE's assurance components. Table 2 provides a listing of all Security Assurance Requirements met by the TOE.

Table 2: Security Assurance Requirements

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

5.2.1 Configuration Management (ACM)

Partial CM automation (ACM_AUT.1)

ACM_AUT.1.1D	The developer shall use a CM system.
ACM_AUT.1.2D	The developer shall provide a CM plan.
ACM_AUT.1.1C	The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.
ACM_AUT.1.2C	The CM system shall provide an automated means to support the generation of the TOE.
ACM_AUT.1.3C	The CM plan shall describe the automated tools used in the CM system.
ACM_AUT.1.4C	The CM plan shall describe how the automated tools are used in the CM system.
ACM_AUT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Generation Support and Acceptance Procedures (ACM_CAP.4)

ACM_CAP.4.1D	The developer shall provide a reference for the TOE.
ACM_CAP.4.2D	The developer shall use a CM system.
ACM_CAP.4.3D	The developer shall provide CM documentation.
ACM_CAP.4.1C	The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.4.2C	The TOE shall be labeled with its reference.
ACM_CAP.4.3C	The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.
ACM_CAP.4.4C	The configuration list shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.4.5C	The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.4.6C	The CM documentation shall describe the method used to uniquely identify the configuration items that comprise the TOE.
ACM_CAP.4.7C	The CM system shall uniquely identify all configuration items that comprise the TOE.
ACM_CAP.4.8C	The CM plan shall describe how the CM system is used.
ACM_CAP.4.9C	The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.
ACM_CAP.4.10C	The CM documentation shall provide evidence that all configuration items have

been and are being effectively maintained under the CM system.

ACM_CAP.4.11C The CM system shall provide measures such that only authorized changes are made to the configuration items.

ACM_CAP.4.12C The CM system shall support the generation of the TOE.

ACM_CAP.4.13C The acceptance plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ACM_CAP.4.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Problem tracking CM coverage (ACM_SCP.2)

ACM_SCP.2.1D The developer shall provide a list of configuration items for the TOE

ACM_SCP.2.1C The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST.

ACM_SCP.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2 Delivery and Operation (ADO)

Detection of modification (ADO_DEL.2)

ADO_DEL.2.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.2.2D The developer shall use the delivery procedures.

ADO_DEL.2.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

ADO_DEL.2.2C The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

ADO_DEL.2.3C The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

ADO_DEL.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence

Installation, generation, and start-up procedures (ADO_IGS.1)

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C	The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.
ADO_IGS.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADO_IGS.1.2E	The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.2.3 Development (ADV)

Fully defined external interfaces (ADV_FSP.2)

ADV_FSP.2.1D	The developer shall provide a functional specification.
ADV_FSP.2.1C	The functional specification shall describe the TSF and its external interfaces using an informal style.
ADV_FSP.2.2C	The functional specification shall be internally consistent.
ADV_FSP.2.3C	The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.
ADV_FSP.2.4C	The functional specification shall completely represent the TSF.
ADV_FSP.2.5C	The functional specification shall include rationale that the TSF is completely represented.
ADV_FSP.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_FSP.2.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

Security enforcing high-level design (ADV_HLD.2)

ADV_HLD.2.1D	The developer shall provide the high-level design of the TSF.
ADV_HLD.2.1C	The presentation of the high-level design shall be informal.
ADV_HLD.2.2C	The high-level design shall be internally consistent.
ADV_HLD.2.3C	The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.2.4C	The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.2.5C	The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or

software.

ADV_HLD.2.6C	The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.2.7C	The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.2.8C	The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_HLD.2.9C	The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.
ADV_HLD.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.2.2E	The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

Subset of the implementation of the TSF (ADV_IMP.1)

ADV_IMP.1.1D	The developer shall provide the implementation representation for a selected subset of the TSF.
ADV_IMP.1.1C	The implementation representation shall unambiguously define the TSF to a level of detail such that the TSF can be generated without further design decisions.
ADV_IMP.1.2C	The implementation representation shall be internally consistent.
ADV_IMP.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_IMP.1.2E	The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

Descriptive low-level design (ADV_LLD.1)

ADV_LLD.1.1D	The developer shall provide the low-level design of the TSF.
ADV_LLD.1.1C	The presentation of the low-level design shall be informal.
ADV_LLD.1.2C	The low-level design shall be internally consistent.
ADV_LLD.1.3C	The low-level design shall describe the TSF in terms of modules.
ADV_LLD.1.4C	The low-level design shall describe the purpose of each module.
ADV_LLD.1.5C	The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

ADV_LLD.1.6C	The low-level design shall describe how each TSP-enforcing function is provided.
ADV_LLD.1.7C	The low-level design shall identify all interfaces to the modules of the TSF.
ADV_LLD.1.8C	The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.
ADV_LLD.1.9C	The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.
ADV_LLD.1.10C	The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.
ADV_LLD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_LLD.1.2E	The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

Informal correspondence demonstration (ADV_RCR.1)

ADV_RCR.1.1D	The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
ADV_RCR.1.1C	For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.
ADV_RCR.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Informal TOE security policy model (ADV_SPM.1)

ADV_SPM.1.1D	The developer shall provide a TSP model.
ADV_SPM.1.2D	The developer shall demonstrate correspondence between the functional specification and the TSP model.
ADV_SPM.1.1C	The TSP model shall be informal.
ADV_SPM.1.2C	The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.
ADV_SPM.1.3C	The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.
ADV_SPM.1.4C	The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

ADV_SPM.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Guidance Documents (AGD)

Administrator Guidance (AGD_ADM.1)

AGD_ADM.1.1D	The developer shall provide administrator guidance addressed to system administrative personnel.
AGD_ADM.1.1C	The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2C	The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3C	The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4C	The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.
AGD_ADM.1.5C	The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.
AGD_ADM.1.6C	The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.
AGD_ADM.1.7C	The administrator guidance shall be consistent with all other documentation supplied for evaluation.
AGD_ADM.1.8C	The administrator guidance shall describe all security requirements on the IT environment that are relevant to the administrator.
AGD_ADM.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence

User Guidance (AGD_USR.1)

AGD_USR.1.1D	The developer shall provide user guidance.
AGD_USR.1.1C	The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.
AGD_USR.1.2C	The user guidance shall describe the use of user-accessible security functions provided by the TOE.
AGD_USR.1.3C	The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.
AGD_USR.1.4C	The user guidance shall clearly present all user responsibilities necessary for secure

operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements on the IT environment that are relevant to the user.

AGD_USR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.5 Life Cycle Support (ALC)

Identification of security measures (ALC_DVS.1)

ALC_DVS.1.1D The developer shall produce development security documentation.

ALC_DVS.1.1C The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.2C The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

ALC_DVS.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2E The evaluator shall confirm that the security measures are being applied.

Basic flaw remediation (ALC_FLR.1)

ALC_FLR.1.1D The developer shall provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.1.1C The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.1.2C The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.1.3C The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.1.4C The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.1.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Developer defined life-cycle model (ALC_LCD.1)

ALC_LCD.1.1D	The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.
ALC_LCD.1.2D	The developer shall provide life-cycle definition documentation.
ALC_LCD.1.1C	The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.
ALC_LCD.1.2C	The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.
ALC_LCD.1.1E	The evaluator shall confirm that the information meets all requirements for content and presentation of evidence.

Well-defined development tools (ALC_TAT.1)

ALC_TAT.1.1D	The developer shall identify the development tools being used for the TOE.
ALC_TAT.1.2D	The developer shall document the selected implementation-dependent options of the development tools.
ALC_TAT.1.1C	All development tools used for implementation shall be well defined.
ALC_TAT.1.2C	The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.
ALC_TAT.1.3C	The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.
ALC_TAT.1.1E	The evaluator shall confirm that the information meets all requirements for content and presentation of evidence.

5.2.6 Security Testing (ATE)

Analysis of coverage (ATE_COV.2)

ATE_COV.2.1D	The developer shall provide an analysis of the test coverage.
ATE_COV.2.1C	The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.
ATE_COV.2.2C	The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.
ATE_COV.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Testing: high-level design (ATE_DPT.1)

ATE_DPT.1.1D	The developer shall provide the analysis of the depth of testing.
ATE_DPT.1.1C	The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.
ATE_DPT.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Functional testing (ATE_FUN.1)

ATE_FUN.1.1D	The developer shall test the TSF and document the results.
ATE_FUN.1.2D	The developer shall provide test documentation.
ATE_FUN.1.1C	The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.
ATE_FUN.1.2C	The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.
ATE_FUN.1.3C	The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.
ATE_FUN.1.4C	The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5C	The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.
ATE_FUN.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

Independent testing – sample (ATE_IND.2)

ATE_IND.2.1D	The developer shall provide the TOE for testing.
ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E	The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE

operates as specified.

ATE_IND.2.3E The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.2.7 Vulnerability Assessment (VLA)

Validation of analysis (AVA_MSU.2)

AVA_MSU.2.1D The developer shall provide guidance documentation.

AVA_MSU.2.2D The developer shall document an analysis of the guidance documentation.

AVA_MSU.2.1C The guidance documentation shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AVA_MSU.2.2C The guidance documentation shall be complete, clear, consistent and reasonable.

AVA_MSU.2.3C The guidance documentation shall list all assumptions about the intended environment.

AVA_MSU.2.4C The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

AVA_MSU.2.5C The analysis documentation shall demonstrate that the guidance documentation is complete.

AVA_MSU.2.1E The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_MSU.2.2E The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

AVA_MSU.2.3E The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

AVA_MSU.2.4E The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

Strength of TOE security function evaluation (AVA_SOF.1)

AVA_SOF.1.1D The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

AVA_SOF.1.1C For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

AVA_SOF.1.2C For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the

specific strength of function metric defined in the PP/ST.

AVA_SOF.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_SOF.1.2E	The evaluator shall confirm that the strength claims are correct.

Independent vulnerability analysis (AVA_VLA.2)

AVA_VLA.2.1D	The developer shall perform a vulnerability analysis.
AVA_VLA.2.2D	The developer shall provide vulnerability analysis documentation.
AVA_VLA.2.1C	The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.
AVA_VLA.2.2C	The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.
AVA_VLA.2.3C	The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.
AVA_VLA.2.4C	The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.
AVA_VLA.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VLA.2.2E	The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.
AVA_VLA.2.3E	The evaluator shall perform an independent vulnerability analysis.
AVA_VLA.2.4E	The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.
AVA_VLA.2.5E	The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low attack potential.

5.3 IT Environment Security Requirements

The IT environment security requirements define functional and/or assurance requirements to be satisfied by the IT environment. The requirements are applicable for hardware, firmware and/or software external to the TOE needed in order to ensure that the security objectives for the TOE are achieved.

FCS_COP.1(f) - Cryptographic operation (asymmetric cryptographic operations)

FCS_COP.1.1(f) ~~The TSF~~ The IT Environment shall perform [***asymmetric key decryption for smart card authentication***] in accordance with a specified cryptographic algorithm [***RSA***] and key sizes [***1024, 2048 and 4096 bits***] that meet the following: [***RSA PKCS#1 version 1.5***].

FPT_STM.1 – Reliable time stamps

FPT_STM.1.1 ~~The TSF~~ The IT Environment shall be able to provide reliable time stamps for ~~its own use~~ the use of the TSF.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Identification and authentication

The TOE requires that users are identified and successfully authenticated before any access is granted to the system and the protected disk partitions. This is performed by the Pre-Boot Environment software which is responsible for unlocking of the encryption keys. In addition, users are required to identify and authenticate before accessing PCMC, and periodically re-authenticate after a period of inactivity of 15 minutes (after which the PCMC display is cleared and locks until the user re-authenticates). The PCMC user identity may be different than that of the pre-boot user, and the PCMC user identity may also be changed from within PCMC using the "Extend Authority" option, requiring re-authentication. ***(FIA_UID.2, FIA_UAU.2), (FTA_SSL.1)***

The TOE employs three different types of authentication mechanisms that can be used to authenticate users: fixed password, smart card authentication and Remote Help authentication. The user's claimed identity is authenticated as described in the paragraphs for each mechanism that follows below. ***(FIA_UAU.5)***

The fixed password authentication mechanism requires that the user enters a password for authentication. Password key derivation uses a PKCS#5 style algorithm where the password is repeatedly one way encrypted using AES, as described in chapter 6.1.5. The derived key is used to decrypt the partition key, for details see chapter 6.1.5. Also see the second last paragraph in the end of this chapter for descriptions over how the TOE detects a failed authentication. The administrator has management tools to adjust the password policy. The administrator can set the minimum length for passwords and determine if passwords may contain spaces or other special characters such as ?, *, ., and &. The administrator can also specify if the password can contain adjacent repetitive, identical characters in a row, set the maximum age of a password and specify the number of passwords that must be used before a previously used password is used again. ***(FIA_SOS.1)***

The TOE's smart card authentication mechanism employs a smart card that stores the user's authentication data. This mechanism requires a smart card reader to check the credentials for access. The smart card stores credentials for the user internally and can only be accessed with a PIN, known only to the owner of the card. In order to perform a successful Smart Card authentication the user has to, by entering the correct PIN-code, unlock the private key corresponding to the public key that was used to encrypt the user key (K_U). The TOE then uses the Smart Card private key to decrypt the user key. The user key, in-turn, decrypts the partition key (see chapter 6.1.5 below).

The TOE's Remote Help authentication mechanisms (one-time login and remote password change) are provided so the TOE administrators can provide login assistance to authorized TOE users under special circumstances, such as a forgotten password or smart card. The type of assistance depends upon the nature of the circumstances. Remote Help is a secondary authentication mechanism, only to be used in special circumstances, not as a normal, everyday, authentication method. The reason for the two different mechanisms relates to when they are used.

> One-time Login is used to provide a temporary login to users who have misplaced or forgotten their smart card. In this case the user will call to the help desk, which will verify the identity of the user using a TOE-independent database (see A.PHONE_DATA). Once verified, the Remote Help flow as described in step 2. below will follow.

Remote Password Change is used to allow the local user to change the TOE password in cases where the current password has been forgotten. In this case the user will call to the help desk, which will verify the identity of the user using a TOE-independent database (see A.PHONE_DATA). Once verified, the Remote Help flow as described in step 2. below will follow.

The Remote Help mechanism is a built in function of the TOE, and does not require setting any parameters except that it can be enabled or disabled on an individual user basis. This includes both the user receiving help, and the account on the computer that is used to provide help. The computer used to provide Remote Help also needs to have a TOE installed on it. To provide Remote Help the administrator accesses the PCMC and uses an account with Remote Help permissions.

Figure 3 shows a conceptual overview of the participants and computations involved in a Remote Help session.

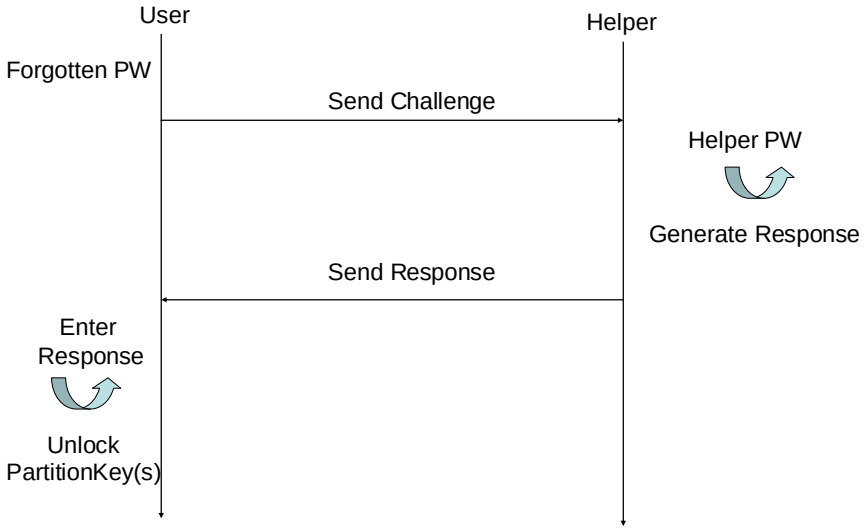

Figure 3 – Remote help overview

The Remote Help flow is:

1. When a user is added, or changes password, on a client the following occurs:

   - A random value is generated, using the FIPS 140-2 approved PRNG. This value will be used as challenge.

   - The Helper's password is one-way encrypted together with the random value. This is the response.

   - The one-way encrypted Helper password encrypts the partition keys.

   - The response is used to encrypt the one-way encrypted Helper's password.

- The encrypted password and the random value used as a challenge is stored in the user database

2. When a user on the client needs Remote Help then:

   - The user gives the Helper his/hers user name
   - The Helper creates a unique value with the Helper user name and the user's user name and presents this to the user
   - The user enters this value and his/her user name
   - The TOE can now find the user's account in the user database and presents the challenge which belongs to the Helper's account
   - The user presents the challenge to the Helper which one-way encrypts it with the Helper's password. The result is the response.
   - The response is read back to the user which enters this in the pre-boot menu and the TOE can decrypt the one-way encrypted Helper's password.
   - When the Helper's one-way encrypted password is unlocked the partition keys can be decrypted
   - Thereafter a new challenge/response pair is generated (see step 1. above)

The remote help algorithm can be described in a more detailed way as follows. The user key (K_U) associated with the remote help account is derived from the password set by the helper using the password derivation mechanism described in chapter 6.1.5. The user key is one-way encrypted together with the challenge using the AES one way encryption process (described in chapter 6.1.5) to create the response. The response is truncated into 12 bytes in order to be manageable in a remote help session. A 12 byte response corresponds to a 96 bit encryption key and is represented as 24 hexadecimal characters or 30 decimal digits when displayed on screen. The challenges are 32- or 64-bit random numbers that are generated using the FIPS 140-2 approved PRNG. The length of the challenge is configurable. The response is padded to 32 bytes, one way encrypted and used to encrypt/decrypt the remote help user key in the same manner as a key directly derived from a password, for details about password based encryption see chapter 6.1.5. Challenges can be generated and the response calculated only when the keys are in non-encrypted state after authentication. The challenge and response are recalculated upon each successful remote help session. Figure 4 shows an overview of generation of the response data..

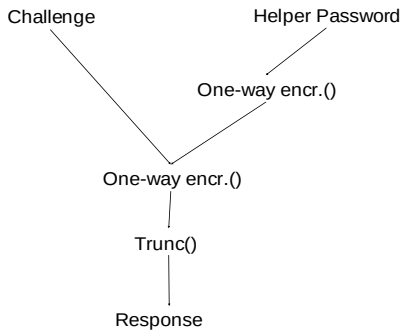

Figure 4 – Generation of Response Data

No authentication data is actually stored or reused specifically for Remote Help, i.e. the user must enter a new password when the encryption keys are unlocked and the user is authorized to access the TOE. ***(FIA_UAU.4)***

Figure 5 depicts an overview of the user database. Information stored in each record in the database include the encrypted partition keys (K_{p1} – K_{pn}) for the encrypted partitions as well as the user identity and the encrypted user key (Ku) and random salt value used for key derivation, see 6.1.5 for details on how keys are encrypted. The disk area where the user database is stored is called the System Area. The entire System Area is encrypted with AES in CBC mode using an internal 256 bit key K_D. This key is stored internally in the program and is used during the startup phase for internal protection only to prevent trivial access to non-sensitive information. The System Area is hidden by the kernel mode components of the TOE. Each user record in the user database stores the user identity along with the encrypted user attributes. ***(FIA_ATD.1)***

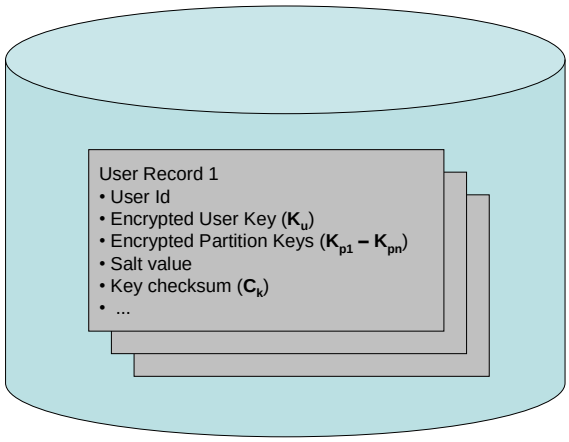

Figure 5: User database overview

In addition to the protection provided by encryption the integrity of the user database is further protected using a checksum over each record. Moreover a checksum is kept for the entire database.

The above checksum is a 32-bit calculated using the following algorithm:

Byte 1: Sum of all bytes being checksummed.

Byte 2: XOR of all bytes.

Byte 3: Multiply and sum of each byte

Byte 4_ Sum of the above

The TOE has the ability to set a policy that locks user accounts when a number of fixed password authentication attempts have failed. When the maximum number of unsuccessful authentication attempts is reached a user variable is set. For each logon attempt this variable is checked. The authorized administrator can via the PCMC set the number of unsuccessful authentication attempts before the user account is locked.

This lock can be absolute, or temporary. A temporary lock specifies a delay period before the user can authenticate again (and again lock if the number of attempts is reached). Even a temporary lock eventually becomes an absolute lock if the user has reached the specified number of attempts. Only authorized administrators are then able to unlock the locked user account from this state. For users authenticating with a smart card, the smart card (outside the TOE) is responsible for locking due to repeated PIN entry failures. ***(FIA_AFL.1)***

For all authentication methods a successful authentication should result in a correctly decrypted or derived user key, K_U, The user key in turn decrypts the partition keys. In order to validate that K_U is correctly derived it is one-way encrypted, using the method described in chapter 6.1.5. Following this the first 8 bytes of the resulting 32 byte blob is XOR:ed into a 4 byte integer value which is compared towards a stored copy (C_k), derived when initializing the user account. If this redundancy check fails the logon process is aborted with an error.

The TOE ensures that user feedback, for all authentication mechanisms, is obscured while the user is entering the password or PIN. ***(FIA_UAU.7)***

Security Functional Requirements: FIA_AFL.1, FIA_ATD.1, FIA_SOS.1, FIA_UAU.2, FIA_UAU.4, FIA_UAU.5, FIA_UAU.7, FIA_UID.2, FTA_SSL.1.

6.1.2 Security Management

The TOE provides the ability for administrators to manage the security functions from the PCMC graphical user interface. These functions include the following:

- User management; creation, removal and modification of user and administrator accounts
- Remote Help assistance; providing users with remote help for special login situations
- Management of security settings
- Audit trail review.

All these functions can be controlled through the assignation of permissions. ***(FMT_SMF.1)***

The TOE uses a hierarchical administrative system with high granularity to allow simplified administration using the inheritance of permissions from higher to lower levels. When installing the TOE two administrator accounts will be created. From these two accounts other groups can be created, and the group settings includes among others the authorization level.

User permissions and other user settings are defined on three levels: User, Group, and Default. Each user is associated with a single user group, e.g. 'System'. For each setting, e.g. 'View Logs', an effective value is computed as follows: if the setting is defined at the User level, then that value is used, unless it is also defined at the Group level with a *Make the group value override the user account value* flag, in which case the Group setting value is used. If the setting is not explicitly defined at the User level, then a Group setting will be used if defined. If no User or Group value has been defined for the setting, then a Default value will be used.

In addition to management restrictions controlled via permission settings, user authorizations are controlled by a Group Authority Level (GAL) from 0 to 9 set for each group. A user can manage other users or groups only if his group's GAL is not lower than that of the managed user or group. In addition, system settings are also associated with a minimum GAL, restricting the user groups that may manage each system setting.

The Administrator's Guide provides a recommended sample user hierarchy that includes: ***(FMT_SMR.1)***

- System Administrators
- Administrators
- Users

The System Administrator is the highest authorization level in the administration of the TOE, assigned all TOE permissions. This role is typically set to perform the following tasks:

- Install the application
- Specify type of protection (boot protection and encryption)
- Identify which drives will be affected
- Create a recovery file (backup of user database)
- Create and administrate profiles for computers (all profiles are protected with a username and password chosen by the creating administrator which are required to re-open the profile)
- Specify user role
- Assign permissions
- Configure system settings, including update validation and log passwords, paths to network file shares, and password composition rules
- Unlock locked accounts
- Add and remove administrators and users, and add, remove, and edit user groups and roles
- Audit trail review

The Administrator role has limited authority in the administration of the TOE. The administrator can typically view logs and provide remote help. Administrators are not allowed to work with users who have higher administration permissions, nor can they raise their own authority level.

The User role has limited authorization to the Pointsec PC application based on what has been defined in the system settings. Each user is assigned an account with a unique user identity and password or PIN that authorizes access to the hard disk. The User role does not have permission to access the PCMC.

This reference hierarchy as described in the Administrator's Guide is used in this ST to express security management restriction requirements. As described above, any user may be assigned individual permissions that allow access to additional functionality, beyond the permissions in this hierarchy. For example, users may be allowed to view local logs on their workstation, but be restricted from reviewing the central log file. In the context of this ST, a user that is allowed to access PCMC is considered to be in an authorized administrator role; if the user also has permissions beyond those of the Administrator Role described above, the user is considered to be in the System Administrator role.

The TOE's administration is designed to allow central control of policy and security settings, but decentralized deployment and day-to-day administration. Through profiles, system administrators are able to install and configure the TOE, delegate authority throughout the network, modify the TOE for local conditions, and assign the properties and authorization of individual users.

The TOE's administration interface provides the utilities for authorized administrators to perform the following tasks as outlined in Table 3 below.

Table 3: Security management PCMC functions

Functions	Function description
User database management	Create, manage and revoke all users and groups and their security relevant attributes including security management roles and permissions and authentication data. **(*FMT_MTD.1(a), FMT_MTD.1(b), FMT_REV.1)***
	Assign authentication mechanisms for each account. ***(FMT_MOF.1)***
	Set the password policy for the fixed password authentication mechanism, including password requirements and allowances. ***(FMT_MOF.1, FMT_MTD.1(c))***
	Set and manage the smart card authentication mechanism properties. ***(FMT_MOF.1)***
	Set the Remote Help authentication mechanism properties (activate one-time login and/or remote password change). ***(FMT_MOF.1)***
	Set and manage the properties for authentication failure handling, such as the number of successive attempts before locking the account. ***(FMT_MTD.1(d), FMT_MTD.2)***
	Set account restrictions such as password expiration time, time limited login or login count expiration. ***(FMT_SAE.1, FMT_MOF.1)***
System settings	Define system settings for clients by specifying what the authorization levels are allowed to do in the system. ***(FMT_MTD.1(a))***
	Specify whether unlocking screensavers and user accounts is allowed. Select whether the client is allowed to receive remote help.
	Identify paths to the locations for update profiles, recovery files, central log files and future program updates.
	Specify whether a new login is required to start the administration program for clients, how long the delay may be when showing audit information at startup, and if Windows' screen saver is allowed.
	Specify whether hibernation is allowed.
	Disable log transfer to Windows Event Log.

Security Functional Requirements: FMT_MOF.1, FMT_MTD.1(a), FMT_MTD.1(b), FMT_MTD.1(c) FMT_MTD.1(d), FMT_MTD.2, FMT_REV.1, FMT_SAE.1, FMT_SMR.1, FMT_SMF.1.

6.1.3 Self-Protection

The TOE implements a specific set of security mechanisms to ensure that other security functions cannot be bypassed and that the security functions themselves cannot be tampered with. During the boot process,

the TOE will not execute if other applications, such as debugging programs, are active, preventing any tampering or spoofing of the login. Within the Windows operating system, the TOE functions as a kernel mode process, restricting access to its execution space and memory. ***(FPT_SEP.1)***

To prevent bypassing of the TSF, the TOE takes control of the Boot Sector of the boot partition, which prevents access to the system without successful authentication. Since the hard disk has been encrypted, authentication to the TOE is the only method for accessing the encryption keys required to access any data. ***(FPT_RVM.1)*** The TOE also employs a suite of self-tests that are performed at system startup and prior to operator login. These self-tests are run automatically when the system is powered on. The boot-code is checked for the presence of debugging tools at each step of the loading process. If suspicious code is detected, the boot process will stop, i.e. the tests must pass before the authentication module becomes operational. The TOE also run cryptographic self-tests on the FIPS 140-2 algorithms. These tests are run at power-up of the TOE and if any of the tests fail the TOE will not load. The cryptographic self-tests applicable for the TOE are:

- Known answer tests:
    - for Triple DES and AES
    - for RSA
    - for the PRNG
- Conditional test: The Cryptographic module performs a continuous random number generator test each time the module produces random data. If a failure of the test is observed, the TOE will immediately enter a disable state. ***(FPT_TST.1)***

Contact with the file share is not necessary for the TOE to continue enforcement of the TSP. No part of the TOE is stored on the server where the file share resides. The TOE has the capability to use redundant servers and will attempt to access another designated distribution server if the communications with the default server should fail. ***(FPT_FLS.1)***

The files that are stored on the file share are encrypted in the same way as when encrypting partition keys in the user database, i.e. the individual values in the files are encrypted using users password or smart card as described in section 6.1.1. This is performed by the TSF before the files are written to the file share. The files are thereby protected from unauthorized disclosure or modification during transit and storage outside of the TOE. The files remain encrypted at all times while outside the TOE. Integrity of audit data is maintained by a HMAC with SHA 256 that is FIPS 140-2 evaluated (Cert. #202). ***(FPT_ITT.1)***

Security Functional Requirements: FPT_FLS.1 , FPT_ITT.1, FPT_RVM.1, FPT_SEP.1, FPT_TST.1.

6.1.4 Auditing

The TOE has the capability to capture an audit record for many different events. Audit is generated as a file on the System Area. The storage of events is circular, and can hold up to 255 event records. As new events are generated then the old events will be overwritten with new events. For each of the events the date/time, type of event, user identity and event outcome is captured in the audit record. The TOE's auditable events include the following: **(*FAU_GEN.1, FAU_GEN.2)***

- All startup events of the computer (auditing starts and stops with the startup and shutdown of the computer)
- All attempts by users to authenticate to the TOE

- All attempts by users to access locked accounts
- Account locked out due to exceeding the maximum number of unsuccessful logon attempts
- All changes to the user database
- Use of the Remote Help feature

The logs are protected from unauthorized access and modification using the encryption/decryption and integrity mechanisms described in chapter 6.1.5.

The TOE provides the ability to transfer the local Pointsec system logs to a central location so they are viewable from one computer. The log entries are transferred to the recovery file location on the server when the computer connects to the network, or once each day. When stored on the central file share the logs are encrypted using 256 bit AES in CBC mode and a log protection key that is derived from an administrator managed log password using the password key derivation mechanism described in chapter 6.1.5. Access control to the logs when they are stored on the central file share is done by using the Windows access rights on the file share. **(*FAU_STG.1*)**

The TOE provides the Log Viewer utility, which is available for the authorized administrator via the PCMC interface for viewing the audit logs on both the local computer and the central file share. Access to this utility is restricted to authorized administrators of the TOE which are required to authenticate themselves before access. The Log Viewer presents the log files in a readable, but read-only, format to the administrator and provides the capability to search and sort through the data using the events category, timestamp, and user identity associated with the event. ***(FAU_SAR.1, FAU_SAR.2, FAU_SAR.3)***

Security Functional Requirements: FAU_GEN.1, FAU_GEN.2, FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_STG.1.

6.1.5 Cryptographic Support

The TOE's Cryptographic Support security function implements several security functions. The cryptographic support mechanisms can be categorized as *cryptographic key management* and *cryptographic operations.* The cryptographic functionality of the TOE is based upon the FIPS 140-2 validated Pointsec Cryptographic Module (FIPS 140-2 certificate #770) embedded in the product. The certificate numbers for the FIPS approved algorithms are HMAC FIPS198 certificate#202, AES FIPS197 certificate#430 and Triple DES FIPS46-3 certificate#459.

The TOE implements the following cryptographic key management functions:

- **Generation and protection:** Each K_P is created during install time by the installation program, and electronically entered into the TOE as part of the install process. It is used during the install process to configure and initially encrypt the partition. The creation of the K_P is internal to the TOE and is generated by a FIPS 140-2 compliant Key Generation algorithm. Partition keys are stored within the user database, which is AES encrypted. No partition keys are stored in the clear. Access can only be obtained to these keys once the user has successfully authenticated to the system. The key K_D is stored internally on the boot partition of the TOE and is used during the startup phase for internal protection only to prevent trivial access to non-sensitive information. ***(FCS_CKM.1)***
- **Zeroize:** To delete partition keys, the System Administrator must remove a user from the user database. That user's respective K_P will then be zeroized. Also, to zeroize all keys, the System Administrator may uninstall the product. ***(FCS_CKM.4)***

The TOE performs the following cryptographic operations:

- **Key encryption and decryption:**

  Password based method - All secret keys are encrypted and stored in the user database. The user selects a username and password at installation time. A key is derived from the password using the one way encryption process described below. The password derived key, in turn, is used to encrypt/decrypt a 256 bit AES key, the user key (K_u). Finally the user key is used to encrypt/decrypt the partition keys (K_{p1} - K_{pn}). See figure 6 for a schematic of key encryption.

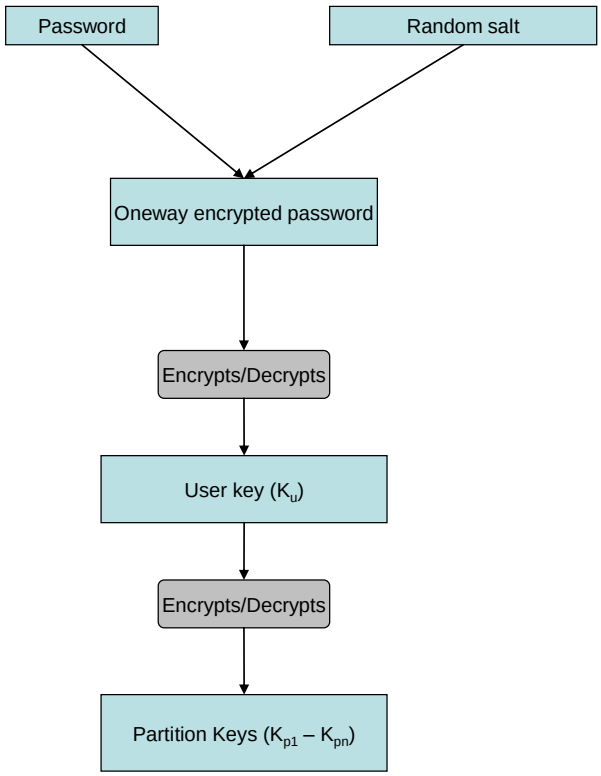

Figure 6 - Schematic of key encryption.

All keys are encrypted using 256 bit AES in ECB mode. In order to get diffusion over the standard 16 byte AES blocks overlapping calls to the ECB encryption are done. The overlapping ECB encryption calls are done on a 4 byte offset basis where, in order to encrypt 32 bytes, a total of 10 calls are done. The first five calls using 4 byte increments of the data input followed by five calls using 4 byte decrements. See Figure 7 for a schematic over the overlapping encryption process. Output from the overlapping calls is the 32 byte encrypted key.

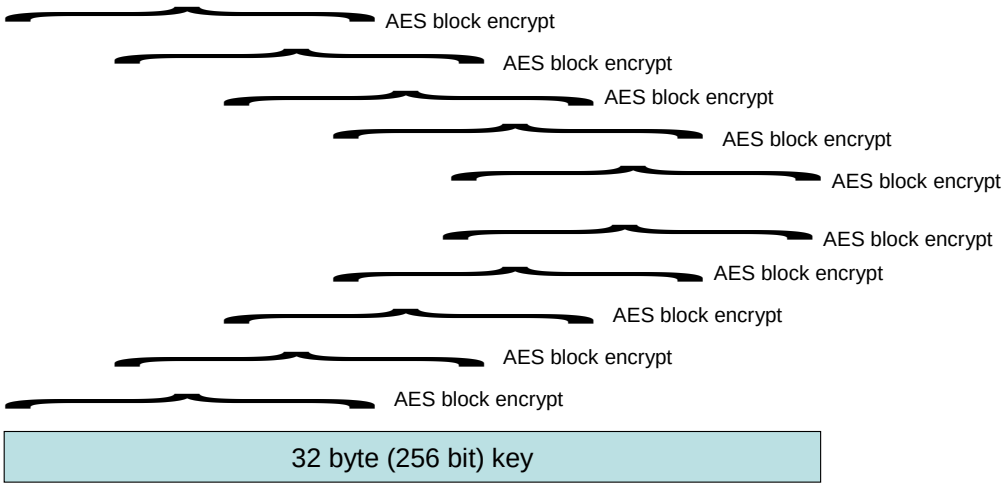

Figure 7 - Schematic for overlapping ECB encryption

Smart card based method - All secret keys are encrypted and stored in the user database. The RSA public key from the user's x.509 certificate is used to encrypt the users' symmetric AES key, which in turn is used to encrypt the partition keys. The keys required to decrypt user data are thereby not stored in the clear. Correct authentication towards the smart card is necessary to access the private key that decrypts the user's symmetric AES key, which in turn decrypts the partition keys (K_{p1} – K_{pn}). AES encryption of keys is done according to the description above for password based key encryption.

Remote Help based method – A remote help account is installed on the machine. The user key associated with the remote help account is derived from the password set by the helper using the password derivation mechanism described above. The user key is one-way encrypted together with the challenge using AES in an irreversible way to create the response. The partition key is in turn AES encrypted together with the remote help user key using the response as key. The challenge and the encrypted keys are stored in the remote help account user database. In order to decrypt the partition key the challenge is sent to the helper, for example over the phone, the helper enters his/her password in order to create the password derived remote help user key, finally the response is created by one-way encrypting the challenge together with the user key. The response is read to the user and the partition key can be decrypted. The challenge and response is recalculated upon a successful remote help session.

(FCS_COP.1(b)), (FCS_COP.1(c)), (FCS_COP.1(f))

- **Data encryption and decryption:** Each partition has a unique key (K_P) that decrypts the hard drive upon successful login. Each partition (or volume) is encrypted with a separate K_P; therefore, each user may have more than one partition key. The K_P is a symmetric encryption key that is used to encrypt all operating system and user files on a partition – everything except the Pointsec system information. Each partition has its own K_P, and thus for a system with two partitions, there would be a K_{P1} and a K_{P2}. The encryption of the disk uses one disk sector as the smallest block (512 bytes). For each block, the relative sector number within the logical volume is used as initialization vector (IV) for the sector encryption. Each sector is encrypted in CBC mode using the selected algorithm. ***(FCS_COP.1(a))***

- **One way encryption:** The AES one-way encryption is run several thousand times in a loop in order to make the key derivation computationally expensive. The basic principle for using 256 bit AES as one-way encryption function for 32 byte (256 bit) data blocks is to encrypt the input data using the data (password) itself as key, i.e. no static key is involved. Moreover for each round in the one-way encryption loop the data is XOR:ed with a random salt, which is regenerated (using

the PRNG) at each successful authentication. The one-way encryption is implemented using AES in ECB mode with standard 256 bit key scheduling. In order to get diffusion over the standard 16 byte AES blocks overlapping calls to the ECB encryption are done. The overlapping ECB encryption calls are done on a 4 byte basis where, in order to encrypt 32 bytes, a total of 10 calls are done. The first five using 4 byte increments of the data input followed by five calls using 4 byte decrements. See Figure 7 for a schematic over the overlapping encryption process. The one-way encryption procedure creates a 32 byte binary blob as output. Reconstructing the original data from the blob is as infeasible as cracking the AES encryption. The resulting blob is used as a key to encrypt and decrypt partition keys (see the key encryption and decryption section above) ***(FCS_COP.1(d))***

- **Integrity of the logs**: Log integrity is ensured by maintaining an HMAC with SHA256 that is FIPS 140-2 approved (Cert #202). The HMAC is stored protected using the log encryption described in section 6.1.4. The key for the HMAC is generated using the FIPS 140-2 approved PRNG and protected using 256 bit AES with a key derived, as described above, from the log password. ***(FCS_COP.1(e))***

A recovery file (a copy of the user database file) can be created for backup purposes if a partition is damaged. The file is stored on a file server and is available only to authorized administrators. To recover encrypted information a single system administrator runs the Recovery Utility from the PCMC. Sensitive data in each individual user-record in the recovery file is protected using the authentication method used by that particular user, i.e. password or smart cards. The recovery file is protected in the same way as the partition key in the user database, i.e. the individual values in the files are encrypted using users password or smart card as described in section 6.1.1. In addition the recovery file is scrambled (AES CBC mode) using static keys (256 bits) in order to avoid trivially exposing semi sensitive information. Integrity of the recovery file is ensured using the same mechanism as for the user database. ***(FCS_COP.1(b)), (FCS_COP.1(c)), (FCS_COP.1(d)***

Security Functional Requirements: FCS_CKM.1 , FCS_CKM.4, FCS_COP.1(a), FCS_COP.1(b), FCS_COP.1(c), FCS_COP.1(d) , FCS_COP.1(e) .

6.1.6 Fault tolerance

Installation files, update profiles, recovery files and software updates can be stored on a file share, which provides member laptops/workstations with a central point for storage. System administrators are able to utilize this to install and configure the system, delegate authorization throughout the network, modify the system for local conditions, and assign the properties and authorization of individual users by using profiles. When a TOE-protected system loses contact with the file share, the TOE provides the administrator with the capability to identify an additional three file servers for redundancy. As a result, if the default server is offline, or the system is unable to contact the server, the system will attempt to communicate with one of the other identified file servers. While a protected system is unable to contact a file server, users are able to continue normal operations and access on the local system. This does not include management functionality, only functionality available to users, such as authentication services. The current profile settings remain in effect until communications can be restored. If contact has been lost to the file share the TSF will continue correct enforcement of the TSP, since the file share is a storage resource that is not a component of the TOE ***(FRU_FLT.1).***

Security Functional Requirement: FRU_FLT.1.

6.1.7 Trusted path

For initial logon, a user must invoke a trusted path in order to ensure the protection of identification and authentication information. The trusted path is invoked by a system reset which is always captured by the TSF (i.e., it cannot be intercepted by an untrusted process). The result will be a logon dialog that is under

the control of the TSF. Once the logon dialog is displayed, the user can enter their identity (username and domain) and authentication (password or PIN-code) ***(FTP_TRP.1).***

Security Functional Requirement: FTP_TRP.1

6.2 TOE Security Assurance Measures

The following assurance measures are applied to satisfy the Common Criteria EAL4, augmented with ALC_FLR.1, assurance requirements.

6.2.1 Process Assurance

6.2.1.1 Configuration Management

The configuration management measures applied by Pointsec ensure that configuration items are uniquely identified, and that documented procedures are used to control and track changes made to the TOE. Pointsec ensures changes to the implementation representation are controlled with the support of automated tools and that TOE associated configuration item modifications are properly controlled. Pointsec performs configuration management on the TOE implementation representation, design, tests, user and administrator guidance, the CM documentation, and security flaws. These activities are documented in the Pointsec PC Configuration Management Manual.

Assurance Requirements: ACM_AUT.1, ACM_CAP.4, ACM_SCP.2.

6.2.1.2 Life-Cycle Support

Pointsec ensures the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan. Pointsec includes security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation necessary to ensure the secure operation of the TOE. Pointsec achieves this through the use of a documented model of the TOE life cycle and well-defined development tools that yield consistent and predictable results. Additionally, Pointsec documents the implementation dependent options and the meaning of all statements used in the implementation. Those procedures and information are documented in the Pointsec PC Life Cycle documentation, which also includes the flaw remediation procedures. The flaw remediation procedures describe how potential security flaws are reported, tracked, analyzed and corrected. The procedures also cover how to provide security flaw information, corrections and guidance to the TOE users.

Assurance Requirements: ALC_DVS.1, ALC_LCD.1, ALC_TAT.1, ALC_FLR.1.

6.2.2 Delivery and Guidance

Pointsec provides delivery documentation and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE and installation and generation instructions at start-up. Pointsec's delivery procedures describe the electronic and non-electronic procedures to be used to detect modification of the TOE. The installation and generation procedures describe the steps necessary to place Pointsec PC into the evaluated configuration. These procedures are documented in the Pointsec PC Delivery and Operation Procedures.

Pointsec provides administrator and user guidance to ensure that the TOE is operated and administered in a secure manner. These documents provide warnings to authorized administrators and users about actions

that can compromise the security of the TOE. Administrator guidance is documented in the Pointsec PC Administrator's Guide, which also includes guidance for the user.

Assurance Requirements: ADO_DEL.2, ADO_IGS.1, AGD_ADM.1, AGD_USR.1.

6.2.3 Design Documentation

An extensive set of design documents has been developed for Pointsec PC to describe all aspects of the TOE security design, architecture, mechanisms, and interfaces. The documents that comprise this set are as follows:

- Pointsec PC Functional Specification: A document that defines the interfaces and functionality of the TOE.
- Pointsec PC High Level Design: A high-level architectural description of the TOE that defines the system through a set of subsystems.
- Pointsec PC Low Level Design: A more detailed representation of the TOE that refines subsystems into modules.
- Pointsec PC Implementation Representation: A representation of the source code used to implement the TOE.
- Pointsec PC Security Policy Model: The Security Policy document fully presents an informal security model for the TOE.
- Pointsec PC Informal Correspondence: A document providing evidence of functional correspondence between the adjacent representations of the TOE. This document will provide a map of all security functions and policies and how they correspond to the design and implementation of the software.

Assurance Requirements: ADV_FSP.2, ADV_HLD.2, ADV_LLD.1, ADV_IMP.1, ADV_SPM.1, ADV_RCR.1.

6.2.4 Tests

The TOE test documentation has been created to demonstrate appropriate breadth and depth of coverage. The test documentation describes how all security relevant interfaces have been tested. The test documentation provides correspondence between the security-relevant interfaces and applicable tests and test variations. The test documentation describes the actual tests, procedures to successfully execute the tests, expected results of the tests, and a set of results from running the tests on the evaluated product.

Assurance Requirements: ATE_COV.2, ATE_DPT.1, ATE_FUN.1, ATE_IND.2.

6.2.5 Vulnerability Assessment

The administrator guidance documentation describes the operation of Pointsec PC and how to maintain a secure state. The administrator guide also describes all operating assumptions and security requirements outside the scope of control of the TOE. The administrator guidance documentation has been developed to serve as a complete, clear, consistent, and reasonable administrator reference. This administrator guidance documentation is documented in the Pointsec PC Administrator's Guide. The Pointsec PC Misuse Analysis document shows that the administrative guidance completely addresses managing the TOE in a secure configuration.

The Strength of TOE Security Function Analysis demonstrates that the SOF claims made in the ST for all probabilistic or permutation mechanisms are correct. Pointsec performs vulnerability analyses of the TOE to identify weaknesses that can be exploited in the TOE. Pointsec has documented the status of identified vulnerabilities and has demonstrated that, for each vulnerability, the vulnerability cannot be exploited in the intended environment and that the TOE is resistant to obvious penetration attacks. The SOF and vulnerability analysis are documented in the Pointsec PC Vulnerability Analysis document

Assurance Requirements: AVA_MSU.2, AVA_SOF.1; AVA_VLA.2.

7. Protection Profile Claims

This TOE does not claim conformance to a Protection Profile.

8. Rationale

This section provides the rationale for completeness and consistency of the Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Functional Requirements;
- Security Assurance Requirements;
- TOE Summary Specification;
- Security Functional Requirement Dependencies; and
- Internal Consistency.

8.1 Security Objectives Rationale

This section show that all threats, secure usage assumptions, and organizational security policies are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat.

8.1.1 Security Objective for the TOE Rationale

Table 4 provides a mapping of TOE security objectives to those threats that the security objectives that the TOE is designed to counter and organizational security policies that the TOE must enforce.

Table 4: Mapping of TOE Security Objectives to Threats or OSPs

TOE Security Objectives	Threats and Organizational Policies
O.AUTHORIZATION	T.ACCESS T.TSF_DATA P.AUTH_USERS T.SYSACC
O.MEDIA_ACCESS	T.SUBVERT T.REMOVE_DISK
O.MANAGE	P.MANAGE
O.AUDIT	T.AUDIT_CORRUPT P.ACCOUNTABILITY

TOE Security Objectives	Threats and Organizational Policies
	T.RECORD_ACTIONS
O.PROTECT	T.TSF_DATA T.UNAUTH_MOD
O.TRUSTED_PATH	T.SPOOF
O.DATA_TRANSFER	T.TRANSIT P.TRANSIT
O.CRYPTO_KEYS	P.CRYPTO_KEYS
O.CRYPTO_OPS	P.CRYPTO_OPS
O.FAULT_TOLERANCE	P.FAULT_TOLERANCE

The following objectives will address the threats and organizational policies listed in the ST.

O.AUTHORIZATION – This objective implements the security policy P.AUTH_USERS, which ensures that only authorized users gain access to the TOE and its resources. Ensuring that only authorized users can access the TOE and its resources counters the threats T.TSF_DATA and T.SYSACC since they require unauthorized access to the TOE. The threat T.ACCESS is also mitigated since only authorized users are granted access to the TOE and any protected resources. Further, access to resources is explicitly granted, preventing an authorized user to from gaining access to other user data.

O.MEDIA_ACCESS – This objective, through the use of whole disk encryption, counters the threats T.SUBVERT and T.REMOVE_DISK. No data can be accessed without successful authentication to decrypt the encryption keys.

O.MANAGE – This objective implements the security policy P.MANAGE by ensuring that only authorized administrators can use the provided utilities for managing the security functions of the TOE and its resources.

O.AUDIT – This objective implements the security policy P.ACCOUNTABILITY, ensuring that all relevant TOE security actions, such as starting and shutting down the TOE, access to the System Area, etc, are recorded in a secure log, which also counters the threat T.RECORD_ACTIONS. This objective counters the threats T.AUDIT_CORRUPT by restricting access to all audit records to only authorized users.

O.PROTECT – This objective counters the threat T.TSF_DATA and T.UNAUTH_MOD by ensuring that internal TOE data can not be accessed by unauthorized users or processes.

O.TRUSTED_PATH – This objective counters the threat T.SPOOF by guaranteeing the user a method of accessing an unmodified and trusted session of the TOE.

O.DATA_TRANSFER – This objective implements the security policy P.TRANSIT, to ensure that TOE internal communications between its distributed parts are protected, countering the threat T.TRANSIT.

O.CRYPTO_KEYS – This objective implements the security policy P.CRYPTO_KEYS which ensures that the cryptographic key operations (generation, destruction and protection) are performed in compliance to FIPS 140-2 (product evaluation) specifications.

O.CRYPTO_OPS – This objective implements the security policy P.CRYPTO_OPS, which ensures that cryptographic operations, such as encryption, used by the TOE to protect all resources, are performed in compliance to FIPS 140-2 (product evaluation), FIPS 46-3 (3DES) and FIPS 197 (AES) specifications as appropriate.

O.FAULT_TOLERANCE – This objective enforces the security policy P.FAULT_TOLERANCE by ensuring that the security functions of the TOE will continue to operate if contact with the file share is lost.

8.1.2 Security Objectives for Environment Rationale

8.1.2.1 Security Objectives for the IT Environment Rationale

Table 5 identifies security objectives for the IT environment in which the TOE is designed to operate and provides a mapping to assumptions that are made about that environment or to applicable threats.

Table 5: Security objectives for the IT environment mapped to assumptions.

TOE Security Objectives for the IT Environment	Assumptions / Threats
OE.TIME_SOURCE	A.TIME
OE.SERVER	A.SERVER
OE.SMART_CARD	T.SYSACC

OE.TIME_SOURCE – The IT environment must provide a reliable time source for the TOE to provide an accurate timestamp for all audit records.

OE.SERVER – The IT environment must provide a file share server to be used as a distribution point for intra-TOE communications, recovery, update and installation files.

OE.SMART_CARD – The IT environment must be able to support O.AUTHORIZATION in authenticating the user by providing a secure token that can perform cryptographic functions that are interoperable with the TOE, to be used by users as an authentication credential.

8.1.2.2 Security Objectives for the Non-IT Environment Rationale

Table 5 identifies security objectives for the non-IT environment in which the TOE is designed to operate and provides a mapping to assumptions that are made about that environment.

Table 6: Security objectives for the non-IT environment mapped to assumptions.

TOE Security Objectives for the Non-IT Environment	Assumptions
OE.MANAGED	A.MANAGE A.NO_EVIL

TOE Security Objectives for the Non-IT Environment	Assumptions
	A.TRAINED_STAFF A.PHONE_DATA
OE.AUTH	A.AUTH_DATA

OE.MANAGED – Ensuring proper installation, management, and operation of the TOE to protect both itself and its resources addresses the assumptions A.MANAGE, A.NO_EVIL, and A.TRAINED_STAFF. This objective ensures that the TOE is operated in a secure manner by competent, trained personnel. It also addresses A.PHONE_DATA by assuring that there is a database containing identification data for users who require login assistance over the phone.

OE.AUTH – Ensuring, through proper user guidance, that TOE user authentication data is kept private addresses the assumption A.AUTH_DATA.

8.2 Security Requirements Rationale

This section provides evidence supporting the combining the internal consistency and completeness of the components (requirements) in the Security Target.

8.2.1 Security Functional Requirements Rationale

Table 7 provides the correspondence mapping between security objectives for the TOE and the security functional requirements that satisfy them.

Table 7: SFRs mapped to Security Objectives

SECURITY FUNCTIONAL REQUIREMENT	O.AUTHORIZATION	O.MEDIA_ACCESS	O.MANAGE	O.AUDIT	O.PROTECT	O.TRUSTED_PATH	O.DATA_TRANFER	O.CRYPTO_KEYS	O.CRYPTO_OPS	O.FAULT_TOLERANCE
FAU_GEN.1				X						
FAU_GEN.2				X						
FAU_SAR.1				X						
FAU_SAR.2				X						
FAU_SAR.3				X						
FAU_STG.1				X						

SECURITY FUNCTIONAL REQUIREMENT	O.AUTHORIZATION	O.MEDIA_ACCESS	O.MANAGE	O.AUDIT	O.PROTECT	O.TRUSTED_PATH	O.DATA_TRANFER	O.CRYPTO_KEYS	O.CRYPTO_OPS	O.FAULT_TOLERANCE
FCS_CKM.1								X	X	
FCS_CKM.4								X	X	
FCS_COP.1(a)		X							X	
FCS_COP.1(b)								X	X	
FCS_COP.1(c)								X	X	
FCS_COP.1(d)								X	X	
FCS_COP.1(e)									X	
FIA_AFL.1	X									
FIA_ATD.1	X									
FIA_SOS.1	X									
FIA_UAU.2	X									
FIA_UAU.4	X									
FIA_UAU.5	X									
FIA_UAU.7	X									
FIA_UID.2	X									
FMT_MOF.1			X							
FMT_MTD.1(a)			X							
FMT_MTD.1(b)			X							
FMT_MTD.1(c)			X							
FMT_MTD.1(d)			X							
FMT_MTD.2			X							

SECURITY FUNCTIONAL REQUIREMENT	O.AUTHORIZATION	O.MEDIA_ACCESS	O.MANAGE	O.AUDIT	O.PROTECT	O.TRUSTED_PATH	O.DATA_TRANFER	O.CRYPTO_KEYS	O.CRYPTO_OPS	O.FAULT_TOLERANCE
FMT_REV.1			X							
FMT_SAE.1			X							
FMT_SMF.1			X							
FMT_SMR.1			X							
FPT_FLS.1					X					
FPT_ITT.1							X			
FPT_RVM.1					X					
FPT_SEP.1					X					
FPT_TST.1					X					
FRU_FLT.1										X
FTA_SSL.1	X									
FTP_TRP.1						X				

In addition to the above table, the environmental requirements, FCS_COP.1(f) and FPT_STM.1, map to the objectives OE.SMART_CARD and OE.TIME_SOURCE, respectively.

O.AUTHORIZATION

FIA_UAU.2 and FIA_UID.2 require a user be authenticated before any access to the TOE and TOE-protected resources is allowed. FTA_SSL.1 locks the PCMC session after a period of user inactivity, requiring the user to re-authenticate.

FIA_ATD.1 and FIA_UAU.5 require that each user be uniquely identified by one of three authentication mechanisms, fixed passwords, smart cards or Remote Help. The unique accounts are then associated with individual attributes for each user.

FIA_UAU.4 prevents the use of previous authentication data that is no longer valid.

FIA_UAU.7 prevents useful feedback from being generated during the entry of a password/PIN/response.

FIA_AFL.1 provides a mechanism for disabling a user account based upon a set of specific conditions, such as a number of failed login attempts.

FIA_SOS.1 provides strength metric for use with fixed passwords.

O.MEDIA_ACCESS

FCS_COP.1(a) require all cryptographic operations, encryption, decryption of data, to be performed in accordance with FIPS 46-3 and FIPS 197 specifications as appropriate. This ensures that all data is protected, controlling access to the stored data.

O.MANAGE

FMT_SMR.1 and FMT_MTD.1(a) require the TOE to provide the ability to set roles for security relevant authority as well as to restrict the ability to define and assign roles to authorized administrators.

FMT_SMF.1 requires that the TOE provide the ability to manage its security functions including the management of security and encryption settings, user management, audit trail review and remote help assistance.

FMT_MOF.1 restricts the ability to enable, disable or modify user identification and authentication data to only authorized administrators.

FMT_MTD.1(b) requires that only authorized users be allowed to modify their authentication data. This includes both users changing their own data as well as authorized administrators.

FMT_MTD.1(c), FMT_MTD.1(d), FMT_MTD.2, FMT_REV.1 and FMT_SAE.1 allow authorized administrators to control the authentication data by placing requirements and limits on acceptable data. These limits could include expiration, password/PIN length, authentication mechanism, revocation, and unsuccessful login attempt consequences.

O.AUDIT

FAU_GEN.1 and FAU_GEN.2 define the TOE events that will be along with the details that will be recorded along with the event.

FAU_SAR.1, FAU_SAR.2, and FAU_SAR.3 restrict access to the audit trail to authorized administrators, and provide them a method for viewing the data according to various criteria.

FAU_STG.1 requires the audit trail to be protected from unauthorized deletion and modification.

O.PROTECT

FPT_TST.1 require the TOE perform a series of internal tests to verify that the integrity of the software has not been compromised.

FPT_FLS.1 provides the TOE will preserve a secure state in the event of a contact failure with the file share or attempts to debug the software at startup.

FPT_SEP.1 ensures the TOE maintains a separate execution domain to protect from external tampering.

FPT_RVM.1ensures that the TOE security policies can not be bypassed.

O.TRUSTED_PATH

FTP_TRP.1 allows the user to gain access to a trusted session of the TOE that is safe from tampering or spoofing.

O.DATA_TRANSFER

FPT_ITT.1 ensures that the TOE will protect all TSF data transferred between distributed parts of the system from disclosure and modification.

O.CRYPTO_KEYS

FCS_CKM.1 and FCS_CKM.4 require all cryptographic keys to be generated, protected, archived, used and deleted in accordance with FIPS 140-2 (product evaluation) specifications.

FCS_COP.1(b) requires all cryptographic operations, encryption, decryption of keys to be performed in accordance with FIPS 46-3 and FIPS 197 specifications as appropriate.

FCS_COP.1(c) requires RSA cryptographic operations for smart card authentication to be performed in accordance with RSA PKCS#1 version 1.5.

FCS_COP.1(d) requires one-way encryption of passwords to be performed using 256 bits AES.

O.CRYPTO_OPS

FCS_COP.1(a), FCS_COP.1(b), FCS_COP.1(c), FCS_COP.1(d), FCS_COP.1(e), FCS_CKM.1 and FCS_CKM.4 require all cryptographic operations, including encryption and decryption of both keys and data, key archiving and key deletion to be performed in accordance with FIPS 140-2 (product evaluation), FIPS 46-3 (3DES), FIPS 197 (AES) and RSA PKCS#1 version 1.5 specifications as appropriate.

O.FAULT_TOLERANCE

FRU_FLT.1 defines that the TOE will continue to enforce all security policies in the event of a contact failure with the file share.

OE.TIME_SOURCE

FPT_STM.1 ensures that an accurate time source will be available to the TOE for use in determining the timestamp for the audit trail.

OE.SMART_CARD

FCS_COP.1(f) ensures that the IT environment can perform asymmetric key decryption for smart card authentication that is interoperable with the TOE.

8.2.2 Security Assurance Requirements Rationale

This ST contains the assurance requirements from the CC EAL4 assurance package, augmented by ALC_FLR.1, and is based on good rigorous commercial development practices. This ST has been developed for a generalized environment with a medium level of risk to the assets.

The TOE will be used to protect attractive information assets and it is assumed that possible attackers will have a medium level of expertise, resources and motivation—an attack potential of medium. The Security Objectives for the TOE were derived to resist attackers with these characteristics, and CC EAL4 augmented with ALC_FLR.1, was found to be sufficient to provide the assurance for the environment.

8.2.3 Requirement Dependency Rationale

Table 8 provides a mapping of security functional requirements and illustrates that all dependencies have been included within this ST, except for the FCS class SFRs' dependency on FMT_MSA.2 (see note below).

Note: FMT_MSA.2 requires that only secure values are accepted in relation to the cryptographic security functional requirements included in this ST. However, the cryptographic mechanisms have been evaluated in accordance with FIPS 140-2 (certificate #770) and as such it is assumed that any requirements for accepting only secure values would have been addressed in that evaluation.

Table 8: SFRs and associated dependencies

Functional Requirements	Dependencies	Dependency	Dependency Met
FAU_GEN.1	FPT_STM.1	FPT_STM.1	✓
FAU_GEN.2	FAU_GEN.1	FAU_GEN.1	✓
	FIA_UID.1	FIA_UID.2	✓
FAU_SAR.1	FAU_GEN.1	FAU_GEN.1	✓
FAU_SAR.2	FAU_SAR.1	FAU_SAR.1	✓
FAU_SAR.3	FAU_SAR.1	FAU_SAR.1	✓
FAU_STG.1	FAU_GEN.1	FAU_GEN.1	✓
FCS_CKM.1	FCS_COP.1	FCS_COP.1	✓
	FCS_CKM.4	FCS_CKM.4	✓
	FMT_MSA.2	-	-
FCS_CKM.4	FDP_ITC.1 or FCS_CKM.1	FCS_CKM.1	✓
	FMT_MSA.2	-	-
FCS_COP.1	FDP_ITC.1 or FCS_CKM.1	FCS_CKM.1	✓
	FCS_CKM.4	FCS_CKM.4	✓
	FMT_MSA.2	-	-
FIA_AFL.1	FIA_UAU.1	FIA_UAU.2	✓
FIA_ATD.1	No dependencies	-	✓
FIA_SOS.1	No dependencies	-	✓
FIA_UAU.2	FIA_UID.1	FIA_UID.2	✓
FIA_UAU.4	No dependencies	-	✓
FIA_UAU.5	No dependencies	-	✓
FIA_UAU.7	FIA_UAU.1	FIA_UAU.2	✓
FIA_UID.2	No dependencies	-	✓
FMT_MOF.1	FMT_SMF.1	FMT_SMF.1	✓

Functional Requirements	Dependencies	Dependency	Dependency Met
	FMT_SMR.1	FMT_SMR.1	✓
FMT_MTD.1	FMT_SMF.1	FMT_SMF.1	✓
	FMT_SMR.1	FMT_SMR.1	✓
FMT_REV.1	FMT_SMR.1	FMT_SMR.1	✓
FMT_SAE.1	FMT_SMR.1	FMT_SMR.1	✓
	FPT_STM.1	FPT_STM.1	✓
FMT_SMF.1	No dependencies	-	✓
FMT_SMR.1	FIA_UID.1	FIA_UID.2	✓
FPT_FLS.1	ADV_SPM.1	CC EAL4	✓
FPT_ITT.1	No dependencies	-	✓
FPT_RVM.1	No dependencies	-	✓
FPT_SEP.1	No dependencies	-	✓
FPT_STM.1	No dependencies	IT Environment	✓
FPT_TST.1	FPT_AMT.1	-	Not met – It is not applicable for the TOE to perform testing to demonstrate the security assumptions made about the underlying abstract machine.
FRU_FLT.1	FPT_FLS.1	FPT_FLS.1	✓
FTA_SSL.1	FIA_UAU.1	FIA_UAU.2	✓
FTP_TRP.1	No dependencies	-	✓
FMT_MTD.2	FMT_MTD.1	FMT_MTD.1	✓
	FMT_SMR.1	FMT_SMR.1	✓

8.2.4 Explicitly Stated Requirements Rationale

This ST does not contain any explicitly stated functional or assurance requirements.

8.2.5 Internal Consistency Rationale

The ST includes no instance of a requirement that contradicts another requirement in the ST. In instances where different requirements apply to the same events or types of data, the requirements and the operations performed within the requirements do not contradict each other, but provide supporting functionality ensuring that the TOE is internally consistent.

The combination of several different supporting security functions, and the inclusion of all dependencies as illustrated in Table 8, ensures that together the selected requirements form a mutually supportive whole. The following items also support this claim:

- mapping and suitability of the requirements to security objectives (as justified in Table 7);
- inclusion of architectural requirements FPT_RVM.1 and FPT_SEP.1 to protect the TSF;

- inclusion of audit requirements to detect attacks of other security functional requirements; and
- inclusion of security management requirements to ensure proper configuration and control of other security functional requirements.

8.2.6 Strength of Function Rationale

The TOE minimum strength of function of SOF-medium was chosen to be consistent with the risk to assets defined within the TOE. The explicit strength of function claim for the authentication mechanism described in FIA_SOS.1 of guessing a fixed password is consistent with the security objectives of the TOE. For fixed passwords the following applies:

- Minimum length of 8 characters
- Digits and letters required
- Mixed case required
- No more than two identical consecutive characters are allowed

Additional users are not allowed to reuse six previous passwords, and are required to change passwords every 90 days. Users are also informed not to use easily guessed passwords.

The SOF-medium claim associated with the key size for the FCS_COP.1 requirement is sufficient to meet the minimum SOF-medium claim of the ST.

The claimed SOF-medium also applies to the Remote Help authentication mechanisms. The Remote Help mechanism is described in chapter 6.1.1.

The SOF-medium strength level is sufficient to meet the objectives of the TOE given the security environment described in the ST.

8.3 TOE Summary Specification Rationale

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions and security assurance measures are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TOE summary specification and indicated in Table 9 are all necessary for the required security functionality in the TSF.

Table 10 provides a mapping of TOE security assurance functions to those security assurance measures that have been implemented by the developer to ensure that the TOE meets the requirements specified by CC EAL4, augmented by ALC_FLR.1.

Table 9: Mapping of SFRs to Security Functions

REQUIREMENT	IDENTIFICATION AND AUTHENTICATION	SECURITY MANAGEMENT	SELF-PROTECTION	AUDITING	CRYPTOGRAPHIC SUPPORT	FAULT TOLERANCE	TRUSTED PATH
FAU_GEN.1				X			
FAU_GEN.2				X			
FAU_SAR.1				X			
FAU_SAR.2				X			
FAU_SAR.3				X			
FAU_STG.1				X			
FCS_CKM.1					X		
FCS_CKM.4					X		
FCS_COP.1(a)					X		
FCS_COP.1(b)					X		
FCS_COP.1(c)					X		
FCS_COP.1(d)					X		
FCS_COP.1(e)					X		
FIA_AFL.1	X						
FIA_ATD.1	X						
FIA_SOS.1	X						
FIA_UAU.2	X						
FIA_UAU.4	X						
FIA_UAU.5	X						
FIA_UAU.7	X						
FIA_UID.2	X						

REQUIREMENT	IDENTIFICATION AND AUTHENTICATION	SECURITY MANAGEMENT	SELF-PROTECTION	AUDITING	CRYPTOGRAPHIC SUPPORT	FAULT TOLERANCE	TRUSTED PATH
FMT_MOF.1		X					
FMT_MTD.1(a)		X					
FMT_MTD.1(b)		X					
FMT_MTD.1(c)		X					
FMT_MTD.1(d)		X					
FMT_MTD.2		X					
FMT_REV.1		X					
FMT_SAE.1		X					
FMT_SMF.1		X					
FMT_SMR.1		X					
FPT_FLS.1			X				
FPT_ITT.1			X				
FPT_RVM.1			X				
FPT_SEP.1			X				
FPT_TST.1			X				
FRU_FLT.1						X	
FTA_SSL.1	X						
FTP_TRP.1							X

Table 10: SARs mapped to Security Assurance Functions

SARs	Process Assurance	Delivery and guidance	Design Documents	Test	Vulnerability Assessment
ACM_AUT.1	X				

SARs	Process Assurance	Delivery and guidance	Design Documents	Test	Vulnerability Assessment
ACM_CAP.4	X				
ACM_SCP.2	X				
ADO_DEL.2		X			
ADO_IGS.1		X			
ADV_FSP.2			X		
ADV_HLD.2			X		
ADV_IMP.1			X		
ADV_LLD.1			X		
ADV_RCR.1			X		
ADV_SPM.1			X		
AGD_ADM.1		X			
AGD_USR.1		X			
ALC_DVS.1	X				
ALC_FLR.1	X				
ALC_LCD.1	X				
ALC_TAT.1	X				
ATE_COV.2				X	
ATE_DPT.1				X	
ATE_FUN.1				X	
ATE_IND.2				X	
AVA_MSU.2					X
AVA_SOF.1					X
AVA_VLA.2					X