Pointsec PC 4.3
Security Target

ST Version 1.08

January 12, 2004

Pointsec Mobile Technologies, Inc.

High Point One
9500 Bormet Drive, Suites 300-302
Mokena, IL 60448

Prepared by:

Science Applications International Corporation

7125 Columbia Gateway Drive, Suite 300
Columbia, MD 21046

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 – Pointsec PC 4.3 Security Target

ST Version – Version 1.08

TOE Identification – Pointsec PC 4.3,

Evaluation Assurance Level (EAL) – EAL 4.

Common Criteria Identification – Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999. International Standard – ISO/IEC 15408:1999.

Keywords – disk encryption, access control, security target, EAL 4, 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 2.1, August 1999, ISO/IEC 15408-2.
    - Part 2 Conformant
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.1, August 1999, ISO/IEC 15408-3.
    - Part 3 Conformant
    - Evaluation Assurance Level 4 (EAL4)

1.3 Strength of Environment

The TOE, Pointsec PC 4.3, has been developed for an operating environment with a moderate level of risk to identified assets. The assurance requirements of EAL 4 and the minimum strength of function of *SOF-medium* were chosen to be consistent with that level of risk.

1.4 Conventions, Terminology, Acronyms

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

1.4.1 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.

1.4.2 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 add, remove, and change settings for specific users.
- **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 by the TOE.
- **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.
- **Disk Partition Access Control SFP:** An access control policy enforced by the TOE that defines rules for controlling access to disk partitions.

- **Dynamic authentication mechanism:** A challenge-response mechanism that supports single-use authentication. The TOE supports any password token that supports the x.9.9 security standard—the standard for Challenge-Response tokens.

- **FIPS 140-1:** 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 TOE requires that passwords contain at least eight characters but no more than 31. The administrator can make changes to the default requirements for passwords.

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

- **One-time Login authentication:** An authentication mechanism whereby a user who normally authenticates with either a smart card or a dynamic token 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.

- **Pointsec Distribution Server:** The central administration server for the TOE. System administrators are able to utilize this server 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.

- **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 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

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

- **User Database Access Control SFP:** An access control policy enforced by the TOE that defines rules for controlling access to the TOE's user database.

1.4.3 Acronyms

The acronyms used within this Security Target:

ACM	Access Control Management
AES	Advanced Encryption Standard
AGD	Administrator Guidance Document
ANSI	American National Standards Institute
ANSI X9.17	A FIPS-compliant PRNG
BS	Boot Sector
CBC	Cipher Block Chaining
CD-ROM	Compact Disk Read Only Memory
CM	Control Management
DAC	Discretionary Access Control
DES	Data Encryption Standard
DO	Delivery Operation
EAL	Evaluation Assurance Level
ECB	Electronic Codebook
FIPS	Federal Information Processing Standards Publication
GB	Gigabyte
I/O	Input/Output
MAC	Message Authentication Code
MBR	Master Boot Record
NIST	National Institute of Standards and Technology
PRNG	Pseudo Random Number Generator
RNG	Random Number Generator (or Generation)
SF	Security Functions
SFR	Security Functional Requirements
SRM	Security Reference Monitor
ST	Security Target
TOE	Target of Evaluation
TSF	Target of Evaluation Security Functions

1.4.4 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 (PPs).
- 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 Type

Pointsec PC 4.3 is a hard disk encryption application that is compliant with FIPS 140-1 Level 1 requirements for cryptographic modules (but not certified). The product is a software based security product for the Windows based PC platform that employs both boot authentication and transparent disk encryption to provide complete protection of information resources stored on fixed media in a workstation, or laptop.

2.2 Product Description

A variety of technologies have been employed to secure PCs and their contents, including physical controls (cables, locks on power supplies, anchored docking stations etc.) and electronic means such as data encryption, user authentication, audit logs and tracking utilities.

Physical access controls are becoming less relevant as users insist on portability. Consequently an increasing emphasis is being made on electronic protection. There are two general types of electronic PC security. The first approach is to provide encryption tools that enable users to protect vital data. This approach, called file encryption, is usually easy to implement but is subject to user discretion regarding what should be secured and the willingness of users to consistently follow the security procedures. Given this dependence on user compliance, organizations seeking an enforceable security program often find file encryption insufficient.

The second approach is much more comprehensive. Here the goal is to prevent unauthorized access to the machine itself, and to provide 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. Because everything on the hard drive is encrypted, this technique is called full hard drive encryption. But that is an oversimplification – strong user authentication and boot protection are necessary components to this complete system.

The importance of boot protection is often misunderstood or confused with the BIOS password schemes offered by the machine manufacturers. Authenticating users before the machine is booted prevents the operating system from being subverted by unauthorized persons using widely available cracking tools. These utilities have proliferated on the Internet and can be used with devastating effect. Unfortunately, most BIOS level protection schemes are fatally weak and cannot be tightly linked with full disk encryption. Boot level access control has the further advantage of providing an effective deterrent to illicit network access via network connected machines, especially if these machines are linked as part of a virtual private network.

While controlling access to the computer is important, this does not by itself protect the data stored on the disk. For example, a simple boot floppy disk could be used to bypass boot protection. Alternatively, removing the drive and placing it in another computer will make the file accessible to brute-force hacking attempts. Even in those rare cases where the drive itself is secured with a password, the data is not encrypted and is therefore vulnerable to several types of attacks. To secure this data, it must be encrypted. Once encrypted, the files will be inaccessible to any unauthorized person.

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.

Figure 1 below illustrates the powerful combination of full-disk encryption and boot protection implemented by Pointsec PC 4.3, as compared to an unprotected system an a system employing file encryption.

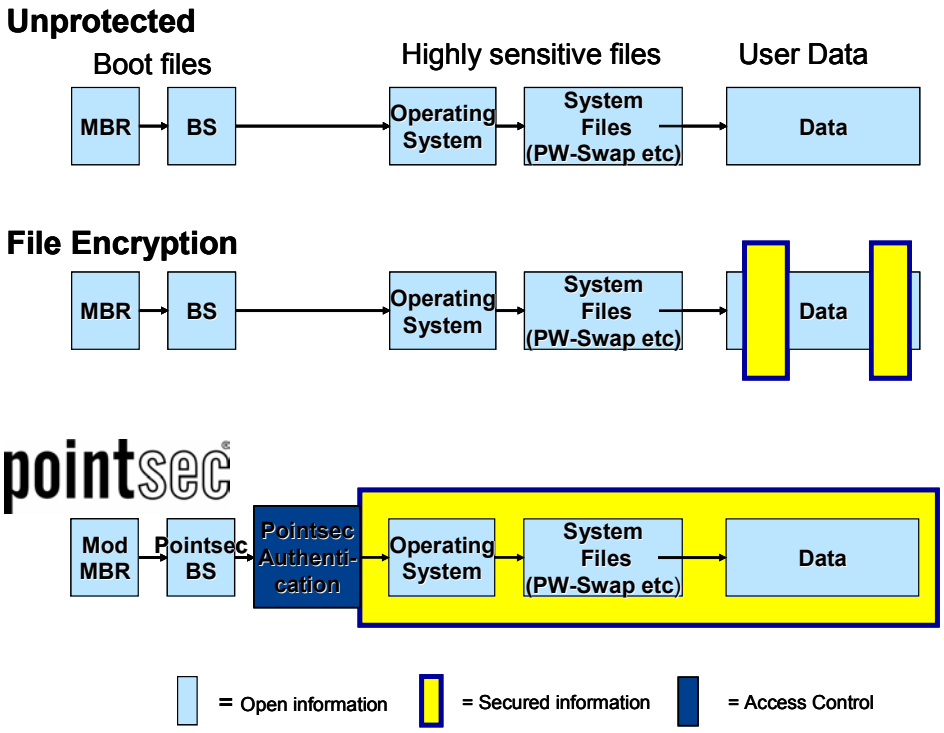

Figure 1: Pointsec PC Full-Disk Encryption

2.3 Product Features

The TOE, Pointsec PC 4.3, is a centrally administered, whole disk encryption and mandatory access control product for use on computers (laptops, desktops or workstations) running Microsoft Windows operating systems. Mandatory access control is provided at the startup of the computer, prior to the loading of the operating system, requiring a successful authentication before the operating system is allowed to boot. Multiple user authentication mechanisms are supported, including fixed passwords, dynamic/challenge-response authentication, smart cards and remote help.

Pointsec PC 4.3 provides 3DES (FIPS 46-3 for 3DES certificate #85) and AES (FIPS 197 certificate #17) as algorithms for whole disk encryption. DES (FIPS 46-3 for DES certificate #146) is used internally for 3DES, but is not available for use the algorithm for whole disk encryption (to FIPS evaluate 3DES, DES must also be evaluated). Pointsec PC 4.3 builds upon the FIPS 140-1 Level 1 certified version Pointsec PC 4.1 (FIPS 140-1 certificate #208). The ongoing product development has maintained the integrity of the code for these certifications as other features and functions have been added and updated, such as including support for new operating systems (Windows XP).

An Administrator performs administration from any client computer. The configuration information is stored in encrypted and signed profiles that are then placed on the Pointsec Distribution Server. These profiles may contain information about authentication mechanisms, authentication data, group memberships, cryptographic configurations and audit configurations.

2.4 Security Environment TOE Boundary

The TOE includes the following physical and logical boundaries.

2.4.1 Logical Boundaries

The Logical Boundaries of the TOE include the interfaces necessary to implement the security policies through the following security functions:

- **Access Control:** Secures desktops and notebooks from unauthorized access, using the combination of boot protection and volume encryption. The TOE enforces access control for each disk partition by employing hard disk encryption, ensuring that unauthorized users are unable to access information on an encrypted device, either from available files, erased files, or temporary files.

- **Identification and authentication:** The TOE provides a flexible suite of five authentication mechanisms, enabling the administrator to assign appropriate authentication requirements for the intended environment. Users authenticate to the system using one of the following mechanisms: fixed password (username/password), smart card authentication (username/smart card/PIN), dynamic authentication (username/token/challenge-response), Remote Help authentication (username/phone identification/TOE challenge/Admin response) and Windows Password Change (username/new password/old password). 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. Windows Password is a special login sequence that occurs when passwords are synchronized between Windows and the TOE. For added security, the computer must be restarted after three consecutive, failed authentication attempts by any authentication mechanism.

- **Security Management:** The TOE provides a number of interfaces to manage the configuration and implementation of the various policies enforced by the TOE. Security management includes managing the following items: authentication data, group memberships, audit data, and cryptographic functions.

- **Self-Protection:** Pointsec PC implements a set of security mechanisms to ensure that other security functions such as access control cannot be bypassed and that the security functions themselves cannot be tampered with. Additionally, mechanisms such as cryptographic self-tests have been implemented to ensure that important cryptographic functions are always operating correctly.

- **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.

- **Cryptographic Support:** The TOE's cryptographic functionality is based upon code that has been certified as meeting the requirements of FIPS 140-1 Level 1. Cryptographic keys are generated, accessed, protected, and destroyed in accordance with requirements defined by FIPS

140-1 Level 1. Additionally, the TOE supports important cryptographic operations such as data and key encryption/decryption.

- **Fault tolerance:** When a Pointsec PC loses contact with the Pointsec Distribution Server, the TOE provides the administrator with the capability to identify an additional three Pointsec Distribution Servers for redundancy. As a result, if the Distribution Server is offline, or the workstation/laptop is unable to contact the Pointsec Distribution Server, the workstation/laptop will attempt to communicate with one of the other identified Distribution Servers.

- **Trusted path:** The TOE provides a mechanism to ensure that users are communicating directly with the TOE during initial authentication. For initial interactive logon, a user must invoke a trusted path in order to ensure the protection of identification and authentication information.

2.4.2 Physical Boundaries

Since the TOE is a software product, its physical boundary is defined by the files and information stored on the computer where it is installed. The TOE can be installed on any x86 compatible computer running Microsoft Windows 2000 and XP (due to the total disk encryption features of the TOE, security patches are not required to be installed for the underlying OS.). Due to the nature of the TOE's security functions, the underlying OS does not prohibit or interfere with the protection provided by the TOE. These functions are implemented uniformly across all listed OS platforms.

The TOE physical boundary consists of:

- The System Area (a protected area on each partition where TOE-specific security information is stored)

- All files within the C:\Program Files\Pointsec directory

  Admin.exe
  Algo.txt
  CrePVR.exe
  DelPatch.exe
  KEYFILE.DIR
  latesttrans.dat
  Log_Lic_adm.exe
  Logo.bmp
  logtrans.dat
  LogTrans.exe
  P95TRAY.exe
  pagentt.exe
  Protect!.lng
  Protect.img
  psearchuser.exe
  PVRLOG.TXT
  ReadmeNt.txt
  recovery.imz
  Roboex32.dll
  splash.bmp
  ssbg.bmp
  Uninst.exe
  UsePVR.exe
  Version.dat

  - Directories

SSO
Update
WORK

- The following files in C:\Winnt\ (or C:\Windows\ as appropriate)

  PMTSEC.DIR

- The following files in C:\Winnt\System32 (or C:\Windows\System32 as appropriate)

  Esso32.dll
  Novpwd32.dll
  Protect!.lng
  Pssonov.ocx
  pagents.exe
  prot_srv.exe
  pscr_nt.scr
  pssocm32.dll
  pssogina.dll
  Config\Pointsec.evt
  Drivers\prot_2k.sys
  Drivers\protmsg.dll

- The following files located in the root directory of each partition

  VOL_CHAR.DAT
  BOOT_SAV.BOT
  PROT_INS.SYS

- Directory (Only XP)

  PrePointsec - System Volume Information

The Pointsec workstation or laptop will also communicate with the Pointsec Distribution Server, as depicted in Figure 2, below. This server provides member workstations/laptops with a central point for storage of installation files, recovery files, update profiles, and software updates. No components of the TOE are installed on the server as all communications are initiated from the workstation. The only requirement for this server is that it be accessible through network communications to the workstation using normal file share access, and not protocols such as ftp or http. Examples of compatible servers include a Windows NT/2000 server, Novell Netware with the Netware client installed locally, a Linux server running SAMBA or a Sun Solaris server with PC-NFS installed on the client. 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).

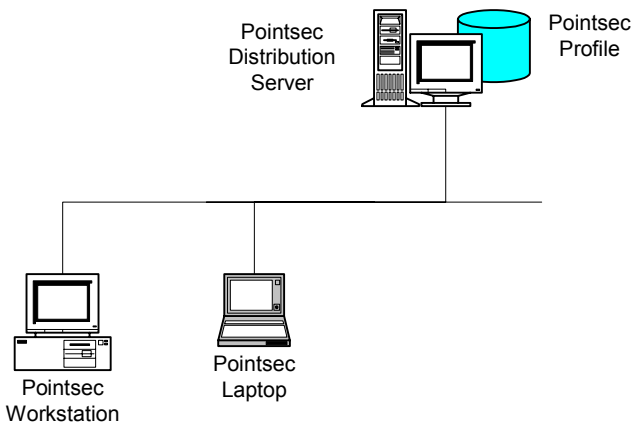

Figure 2: TOE's Physical Environment

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 developed 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 separate parts of the TOE 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.OBJECT_REUSE	Users may request access to resources and gain unauthorized access to information by using out-of-date authentication data, thereby releasing information to the subsequent user.
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 140-1 Level 1 (product evaluation).

P.CRYPTO_OPS	All cryptographic operation performed by the system will be compliant the requirements of FIPS 140-1 (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 access control functions continue to operate if systems lose communications with central administration servers.

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 Physical Assumptions

A.TIME	The TOE's IT environment will provide a reliable time source to enable the TOE to timestamp audit records.
A.SERVER	The TOE's IT environment will provide a distribution server for the management of the TOE client software. This server provides installed PCs with a central point for storage of installation files, recovery files, update profiles, and software updates.

3.3.3 System 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.ACCESS_CONTROL	The TSF must control access to each logical partition based on identity of users. The TSF must provide the ability to limit each user's access.
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 TOE
O.CRYPTO_KEYS	The TSF must ensure that cryptographic keys are generated, accessed, protected, and destroyed in accordance with requirements defined by FIPS 140-1 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 140-1 Level 1 (product evaluation), FIPS 46-3 (3DES) and FIPS 197 (AES).
O.RESIDUAL_INFO	The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.

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

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 server to be used as a distribution point for intra-TOE communications, log, recovery, update and installation files.

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 other authentication information, 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.3 - Cryptographic key access	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
User Data Protection (FDP)	FDP_ACC.2(a) - Complete access control (Disk Partition Access Control SFP)	Iteration, Assignment
	FDP_ACC.2(b) - Complete access control (User Database Access Control SFP)	Iteration, Assignment
	FDP_ACF.1(a) - Security attribute based access control (Disk Partition Access Control SFP)	Iteration, Assignment
	FDP_ACF.1(b) - Security attribute based access control (User Database Access Control SFP)	Iteration, Assignment

Functional Class	Functional Components	ST Operation
	FDP_RIP.1 - Subset residual information protection	Selection, 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
	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(a) - Management of security functions behavior (Audit)	Selection, Assignment, Iteration
	FMT_MOF.1(b) - Management of security functions behavior (Identification and authentication)	Selection, Assignment, Iteration
	FMT_MOF.1(c) - Management of security functions behavior (Cryptographic support)	Selection, Assignment, Iteration
	FMT_MSA.1 - Management of security attributes	Assignment, Selection
	FMT_MSA.2 - Secure security attributes	None
	FMT_MSA.3 - Static attribute isolation	Assignment, Selection
	FMT_MTD.1(a) - Management of TSF Data (audit trail)	Selection, Assignment, Iteration
	FMT_MTD.1(b) - Management of TSF Data (security-relevant roles)	Selection, Assignment, Iteration

Functional Class	Functional Components	ST Operation
	FMT_MTD.1(c) - Management of TSF Data (partition keys)	Selection, Assignment, Iteration
	FMT_MTD.1(d) - Management of TSF Data (authentication data)	Selection, Assignment, Iteration
	FMT_MTD.1(e) - Management of TSF Data (password policy)	Selection, Assignment, Iteration
	FMT_MTD.1(f) - 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 TSF (FPT)	FPT_AMT.1 – Abstract machine testing	Selection
	FPT_FLS.1 – Failure with preservation of secure state	Assignment
	FPT_ITT.1 - Basic internal TSF data transfer protection	Selection
	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
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; and

c) [***All attempts to use authentication mechanisms***

d) ***All attempts to access locked user accounts***

e) ***All changes to a user's authentication data***

f) ***All attempts to uninstall the TOE from the system***

g) ***Account locked out due to exceeding the maximum number of unsuccessful logon attempts***

h) ***Changes to the system time***

i) ***All changes to the user database***

j) ***Use of the Remote Help feature***]

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 **(if applicable)**[1], 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***].[2]

FAU_GEN.2 - User identity association

FAU_GEN.2.1 **For audit events resulting from actions of identified users,** ~~The~~ **the** TSF shall be able to associate each auditable event with the identity of the user that caused the event.[3]

[1] This requirement has been modified to comply with U.S Interpretation #410.
[2] This requirement has been added to comply with U.S Interpretation #410.
[3] This requirement has been modified to comply with U.S Interpretation #410.

FAU_SAR.1 - Audit review

FAU_SAR.1.1 The TSF shall provide [***authorized users***] with the capability to read [***all audit information (to administrators) and local computer information (authorized users)***] 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) and 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 **in the audit trail** from unauthorized deletion. [4]

FAU_STG.1.2 The TSF shall be able to [*prevent*] modifications to the audit records **in the audit trail**.[5]

5.1.1 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 140-1 Compliant Key Generation Algorithm***]

and specified cryptographic key sizes [

a) ***168 bits (3DES)***

b) ***128, 192 and 256 bits (AES)***]

that meet the following: [***FIPS 140-1 Level 1, Section 4.8.1 Key Generation***].

[4] This requirement has been added to comply with U.S Interpretation #422.
[5] This requirement has been added to comply with U.S Interpretations #422 and #423.

FCS_CKM.3 - Cryptographic key access

FCS_CKM.3.1 The TSF shall perform [***cryptographic key archival***] in accordance with a specified cryptographic key access method [***duplicate***] that meets the following: [***FIPS 140-1 Level 1, Section 4.8.6 Key Archiving***].

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 140-1 Level 1, Section 4.8.5 Key Destruction***].

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 [***Triple DES or AES***] and key sizes [***168 bits (Triple DES) and 128 192 and 256 bits (AES)***] that meet the following: [***FIPS 46-3 (3DES) and FIPS 197 (AES)***].

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 [***Triple DES or AES***] and key sizes [***168 bits (Triple DES) and 128 192 and 256 bits (AES)***] that meet the following: [***FIPS 46-3 (3DES) and FIPS 197 (AES)***].

5.1.2 User Data Protection (FDP)

FDP_ACC.2(a) - Complete access control (Disk Partition Access Control SFP)

FDP_ACC.2.1(a) The TSF shall enforce the [***Disk Partition Access Control SFP***] on [***subjects: authorized users and objects: disk partitions***] and all operations among subjects and objects covered by the SFP.

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

FDP_ACC.2(b) - Complete access control (User Database Access Control SFP)

FDP_ACC.2.1(b) The TSF shall enforce the [***User Database Access Control SFP***] on [***subjects: authorized users and objects: user database***] and all operations among subjects and objects covered by the SFP.

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

FDP_ACF.1(a) - Security attribute based access control (Disk Partition Access Control SFP)

FDP_ACF.1.1(a) The TSF shall enforce the [***Disk Partition Access Control SFP***] to objects based on [***user identity and partition key(s) (K_p)***].

FDP_ACF.1.2(a) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

a) ***An authorized user will be granted access to a disk partition if their user identity is associated with a partition key that decrypts the disk partition***].

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

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

FDP_ACF.1(b) - Security attribute based access control (User Database Access Control SFP)

FDP_ACF.1.1(b) The TSF shall enforce the [***User Database Access Control SFP***] to objects based on [***security-relevant role and privileges***].

FDP_ACF.1.2(b) The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [

a) ***Authorized users will be granted access, and allowed to perform specific operations, on the user database according to privileges assigned to their security-relevant role***].

FDP_ACF.1.3(b) The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [

a) ***Authenticated users will be granted access to the user database to modify their own authentication data***].

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

FDP_RIP.1 - Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*deallocation of the resource from*] the following objects: [***user database***].

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 [***any 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-relevant roles;***

c) ***Privileges;***

d) ***Authentication data; and***

e) ***Partition key(s) ($K_{p)}$***]

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***].

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 [***the Dynamic Authentication Mechanism, and Remote Help Authentication Mechanism***].

FIA_UAU.5 - Multiple authentication mechanisms

FIA_UAU.5.1 The TSF shall provide [

a) ***Fixed Password Mechanism***

b) ***Dynamic Authentication Mechanism***

c) ***Smart card Authentication Mechanism***

d) ***Remote Help Authentication Mechanism***

e) ***Windows Password Change 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(a) - Management of security functions behavior (Audit)

FMT_MOF.1.1(a) The TSF shall restrict the ability to [*enable, disable, modify the behavior of*] the function [***audit***] to [***authorized administrators***].

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

FMT_MOF.1.1(b) The TSF shall restrict the ability to [*enable, disable, modify the behavior of*] the function [***identification and authentication***] to [***authorized administrators***].

FMT_MOF.1(c) - Management of security functions behavior (Cryptographic support)

FMT_MOF.1.1(c) The TSF shall restrict the ability to [*enable, disable, modify the behavior of*] the function [***cryptographic support***] to [***authorized administrators***].

FMT_MSA.1 – Management of security attributes

FMT_MSA.1.1 The TSF shall enforce the [***User Database Access Control SFP and Disk Partition Access Control SFP***] to restrict the ability to [*modify, delete, and* ***create***] the security attributes [***user identifier, security-relevant roles, privileges, authentication data, and partition key(s)***] to [***authorized administrators***].

FMT_MSA.2 – Secure security attributes

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

FMT_MSA.3 - Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the [***User Database Access Control SFP and Disk Partition Access Control SFP***] to provide [*restrictive*] default values for security attributes that are used to enforce the SFP.

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

FMT_MTD.1(a) - Management of TSF Data (audit trail)

FMT_MTD.1.1(a) The TSF shall restrict the ability to [*query, and* ***export***] the [***audit trail***] to [***authorized administrators***].

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

FMT_MTD.1.1(b) The TSF shall restrict the ability to [*modify*] the [***security-relevant roles for users***] to [***authorized administrators***].

FMT_MTD.1(c) - Management of TSF Data (partition keys)

FMT_MTD.1.1(c) The TSF shall restrict the ability to [***initialize****, delete*] the [***partition keys***] to [***authorized administrators***].

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

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

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

FMT_MTD.1.1(e) 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 or dynamic authentication)***] to [***authorized administrators***].

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

FMT_MTD.1.1(f) The TSF shall restrict the ability to [*modify*] the [***settings for handling authentication failures***] to [***authorized administrators***].

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 [***authorized administrators***].

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 an authorized 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 [***authorized administrators***].

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 [***authorized administrators***].

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[6]

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, including encryption***

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.

5.1.5 Protection of TSF (FPT)

FPT_AMT.1 – Abstract machine testing

FPT_AMT.1.1 The TSF shall run a suite of tests [*during initial start-up*] to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.

FPT_FLS.1 – Failure with preservation of secure state

FPT_FLS.1.1 The TSF shall preserve a secure state when the following types of failures occur: [***communications are lost with the Pointsec Distribution Server***].

FPT_ITT.1 - Basic internal TSF data transfer protection

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

[6] This requirement has been added to comply with International Interpretation #65

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 and access***] when the following failures occur: [***communications are lost with the Pointsec Distribution Server***].

5.1.7 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. 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_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.
International Interpretation RI #3	The configuration list shall uniquely identify all configuration items that comprise the TOE.[7]
ACM_CAP.4.4C	The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.4.5C	The CM documentation shall describe the method used to uniquely identify the configuration items.
ACM_CAP.4.6C	The CM system shall uniquely identify all configuration items.
ACM_CAP.4.7C	The CM plan shall describe how the CM system is used.
ACM_CAP.4.8C	The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

[7] This requirement has been added to comply with International Interpretation #3

ACM_CAP.4.9C	The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.
ACM_CAP.4.10C	The CM system shall provide measures such that only authorized changes are made to the configuration items.
ACM_CAP.4.11C	The CM system shall support the generation of the TOE.
ACM_CAP.4.12C	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 CM documentation~~. **The developer shall provide a list of configuration items for the TOE**[8]
ACM_SCP.2.1C	~~The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.~~ 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.[9]
~~ACM_SCP.2.2C~~	~~The CM documentation shall describe how configuration items are tracked by the CM system.~~[10]
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

[8] This requirement has been modified to comply with International Interpretation #4
[9] This requirement has been modified to comply with International Interpretation #4
[10] This requirement has been deleted to comply with International Interpretation #4

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 documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.~~

The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE.[11]

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

[11] This requirement has been modified to comply with International Interpretation #51

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.

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 and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.

AVA_VLA.2.2D The developer shall document the disposition of identified vulnerabilities.

AVA_VLA.2.1C ~~The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.~~

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.[12]

AVA_VLA.2.2C ~~The documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.~~

The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.[13]

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.[14]

AVA_VLA.2.4C The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.[15]

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.

[12] This requirement has been modified to comply with International Interpretation #51
[13] This requirement has been modified to comply with International Interpretation #51
[14] This requirement has been added to comply with International Interpretation #51
[15] This requirement has been added to comply with International Interpretation #51

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 satisfied hardware, firmware and/or software external to the TOE needed in order to ensure that the security objectives for the TOE are achieved.

FPT_STM.1 – Reliable time stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps for its own use.

6. TOE Summary Specification

This chapter describes the security functions and associated assurance measures.

6.1 TOE Security Functions

6.1.1 Access Control

The TOE enforces two access control policies: the *Disk Partition Access Control SFP and* the *User Database Access Control SFP*. Each of these access control policies have a unique set of subjects and objects and defines a set of controlled operations between the identified subjects and objects. Disk encryption, combined with strong user authentication, provides the TOE with the capability to enforce an access control policy ensuring that only authorized users are able to access information for which they have access rights.

The TOE enforces the *Disk Partition Access Control SFP* by ensuring that only authorized users are granted access to a specific disk partition. Each partition is encrypted with a separate K_P; therefore, multiple partition keys could be associated with each user. The K_P is a symmetric encryption key used to encrypt all operating system and user files on a partition– everything except the Pointsec System Area, which is encrypted with K_D (a symmetric encryption key) on each partition. 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} for each partition, but K_D is the same for all System Areas.

The System Area consists of two parts: the boot-strapping software and the user database. The boot-strapping software and user attributes are stored only on the boot partition. The user database is split into pieces based upon the number of partitions; each partition stores its own copy of the database, containing only the user accounts that have access to that partition.[16] ***(FDP_ACC.2(a), FDP_ACF.1(a))***

The TOE also enforces the *User Database Access Control SFP* protecting the user database from access by unauthorized users. All secret and private keys are encrypted and stored in the user database along with other important TSF data. The various security-relevant roles can be assigned privileges to allow specific access to the user database. The privileges that can be assigned to an authorized administrator include the following: ***(FDP_ACC.2(b), FDP_ACF.1(b))***

- Change Properties for Users
- Change Properties for Administrators
- Add Users
- Add Groups
- Administer Profiles
- Delete Users
- Delete Groups
- Delete Profiles

[16] When using the term user database, it includes all pieces on all encrypted partitions treated as one.

When a user is entered into or deleted from the database, the database file is reconstructed with the new set of user records, AES-CBC encrypted with a key K_D, and rewritten to the hard drive. This results in temporary multiple copies of the user database file, including the old file, and backup copy, and the new version of the file. The old database files are zeroized before being overwritten. Once the new database files (main copy and backup copy) are in place, the temporary database is also zeroized. ***(FDP_RIP.1)***

Security Functional Requirements: FDP_ACC.2(a), FDP_ACC.2(b), FDP_ACF.1(a), FDP_ACF.1(b), FDP_RIP.1.

6.1.2 Identification and authentication

The TOE requires that all users be identified and successfully authenticated before any access is granted to the system and the protected disk partitions. ***(FIA_UID.2, FIA_UAU.2)*** The TOE employs five different types of authentication mechanisms that can be used to authenticate users: fixed password, dynamic/challenge-response authentication, smart card authentication, Remote Help authentication and Windows Password Change authentication. To enhance the security of the authentication process, the computer must be restarted after three consecutive, failed authentication attempts, regardless of which authentication mechanism is used. ***(FIA_UAU.5)***

The fixed password authentication mechanism requires that the user enter a password for authentication. The password must contain at least eight characters but no more than 31. The administrator has management tools to adjust the password policy. The administrator can 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. ***(FIA_SOS.1)***

The TOE's Dynamic authentication mechanism utilizes a dynamic token that generates a one-time passwords based upon a challenge-response prompting mechanism. The computer first issues a challenge, which the user inputs into the token. The token then issues a response that is typed back into the computer. A match will allow access. Dynamic tokens are linked to a single user. When the account is created the encryption key for the token is programmed into the TOE.

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. The TOE supports PKCS #11 enabled smart cards, such as ActivCard Gold, Datakey and TeliaID smart cards.

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, token 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 names relates to when they are used.

> One-time Login is used to provide a temporary login to users who have misplaced or forgotten either their smart card or dynamic token. 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 user will read a challenge generated by the TOE to the administrator. The administrator will then enter this along with other login information to generate a response. This response is read back to the user and entered into the TOE. The user is then authorized to access the TOE for the duration of this session. Upon a restart of the TOE, the authorization expires.
>
> Remote Password Change is used to allow the local user to change the TOE password in cases where the current password has been lost or 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 user will read a challenge generated by the TOE to the administrator. The administrator will then enter this along with other login information to generate a response. This response is read back to the user and entered into the TOE. The user is then authorized to access the TOE and immediately required to enter a new password.

The Remote Help mechanism is a built in function of the TOE, and does not require setting any parameters except that it be enabled for the user. This includes both the user receiving help, and the account on the computer that is used to provide help. The challenge generated on the client is a randomly generated string that is then combined with user information. The string is read on the helping system and a response is generated that also includes login information about the administrator providing the help. No authentication data is actually stored specifically for Remote Help.

The TOE's Windows Password Change mechanism is used to verify the new TOE password after a change has been initiated from within Windows (using Windows mechanisms for password change) using the TOE password synchronization feature. This authentication process allows the TOE to verify that the password change is authentic by requiring that the user first enter the new password (the one entered within Windows), followed by the previous TOE password. If the user is unable to successfully enter the previous TOE password, the user will not be successfully authenticated and the synchronized password is deleted. Successful entry of both passwords will change the TOE password to the synchronized password and successfully authenticate the user.

The TOE prevents the reuse of authentication data by zeroization of previous data (see section 6.1.6 for more information on zeroization). ***(FIA_UAU.4)***

The disk encryption keys in the user database are AES encrypted with a symmetric key derived from the user password K_U and stored along with each individual user record as depicted in Figures 3 and 4 below. The operator selects a username and password at installation time. The password is then securely hashed using a one-way process that includes an array of 10 times AES used in an irreversible way, and the result of the hash is the key used to encrypt the partition key. The hash itself is not stored on the disk. Each successful logon with correct username and password recreates the hash to decrypt the key for the partition. The keys required to decrypt user data are not stored in the clear. Correct authentication of a user will generate a unique hash to decrypt the user's partition key. When a user is first created, the authentication data entered by the authorized administrator is used to create this hash.

Figure 3 depicts the structure of the user database on the boot partition, which includes both the partition key for the boot partition as well as the user attributes. The entire System Area is encrypted with an internal key K_D. This key is stored internally and used for internal protection only. As shown in Figure 4, all other encrypted partitions only store the partition key, not any user attributes. The following attributes are stored for each user: user identity, authentication data, security roles, privileges and encryption keys. ***(FIA_ATD.1)***

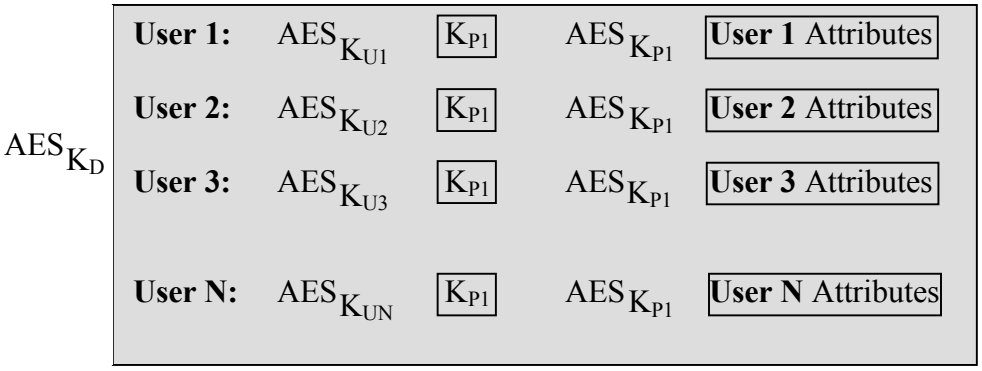

Figure 3: Encrypted user database on the Boot Partition

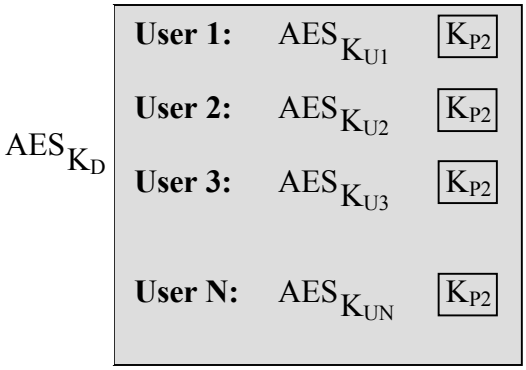

Figure 4: Encrypted user database on a non-Boot Partition

The TOE has the ability to set a policy that locks user accounts when a number of authentication attempts have failed. The User Database Management tool, within the administration interface, allows the authorized administrator to 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. ***(FIA_AFL.1)***

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.

6.1.3 Security Management

The TOE provides the ability to manage the security functions of the software. These functions include the following:

- User management, the creation, removal and modification of user accounts, whether users or administrators.
- Remote Help assistance, providing users with remote help for special login situations
- Management of security settings, including which media will be encrypted
- Audit trail review.

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

The TOE uses a hierarchical system to allow simplified administration using the inheritance of permissions from higher to lower levels. The application supports three specific security relevant roles: ***(FMT_SMR.1.)***

- System Administrators
- Administrators
- Users

The System Administrator is the highest authorization level in the administration of the TOE. This role can perform the following tasks:

- Install the application
- Specify type of protection (boot protection and encryption, encryption, etc.)
- 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)
- Generate new partition keys for added users
- Specify user role
- Assign privileges
- Configure system settings
- Unlock locked accounts
- Add and remove administrators and users

The Administrators have limited authority in the administration of the TOE. The administrator can add, remove, and change settings for specific users. Administrators are not allowed to work with users who have higher administration privileges, nor can they raise their own authority level. Administrators are normally given the authorization to provide remote assistance and to modify profiles.

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 that authorizes access to the entire hard disk or only specific partition(s) on the hard disk.

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 utilities

Utilities	Function description
User database management	Create and manage all users and groups and their security relevant attributes including user identity, security-relevant roles, privileges, authentication data and partition keys. ***(FMT_MSA.1, FMT_MTD.1(d,), FMT_REV.1).***
	Assign and disable authentication mechanisms for each account. ***(FMT_MOF.1(b))***

Utilities	Function description
	Set the password policy for the fixed password authentication mechanism, including password requirements and allowances. ***(FMT_MOF.1(b), FMT_MTD.1(e))***
	Set and manage the dynamic authentication mechanism properties. ***(FMT_MOF.1(b))***
	Set and manage the smart card authentication mechanism properties. ***(FMT_MOF.1(b))***
	Set the Remote Help authentication mechanism properties (activate one-time login and/or remote password change). ***(FMT_MOF.1(b))***
	Set and manage the properties for authentication failure handling, such as the number of successive attempts before locking the account. ***(FMT_MTD.1(f), FMT_MTD.2)***
	Set account restrictions such as password expiration time, time limited login or login count expiration. ***(FMT_SAE.1, FMT_MOF.1(b))***
Profile creation and management	The Protection (or Profile) function enables the authorized administrator to specify which drives, volumes are to be registered and the type of security and algorithm they are to be assigned. Providing the capability to enable, disable, and modify the behavior of the cryptographic functionality of the TOE. ***(FMT_MOF.1(c), FMT_MTD.1(c))***
System settings	Define system settings for clients by specifying what the three authorization levels are allowed to do in the system. ***(FMT_MTD.1(b))***
	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. ***(FMT_MOF.1(a))***
	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 Windows Password Change is enabled.
	Disable log transfer to Windows NT Event Log.
Audit Log Review	Enables the administrator to view the audit logs. Provides the functionality to select, sort and export the various audit records. ***(FMT_MTD.1(a))***
	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.

By default, new users are not provided with partition keys. The authorized administrator must assign disk partition keys to a user. ***(FMT_MSA.3)*** Additionally, The TOE restricts the data entry values for each of the following security attributes: ***(FMT_MSA.2)***

- **User identity:** The TOE ensures that only unique users can be entered in the user database.
- **Authentication data:** The TOE restricts the allowed authentication data according to the policy set by the various authentication mechanisms.
- **Security-relevant role:** The TOE has three defined roles: User, Administrator, and System Administrator. The TOE will not accept any other value.
- **Privileges:** The TOE has a defined set of privileges with associated operations that can be performed on the user database.
- **Partition Keys:** The TOE uses the FIPS 140-1 (product evaluation) approved cryptographic module to ensure that only secure values can be created for this security attribute.

Security Functional Requirements: FMT_MOF.1(a), FMT_MOF.1(b), FMT_MOF.1(c), FMT_MSA.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1(a), FMT_MTD.1(b), FMT_MTD.1(c), FMT_MTD.1(d), FMT_MTD.1(e) FMT_MTD.1(f), FMT_MTD.2, FMT_REV.1, FMT_SAE.1, FMT_SMR.1, FMT_SMF.1.

6.1.4 Self-Protection

The TOE implements a specific set of security mechanisms to ensure that other security functions such as access control 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 after the system BIOS tests. The tests must pass before the authentication module becomes operational. The TOE runs AES-MAC, an integrity check, to ensure that no malicious code has been loaded and will perform cryptographic known answer tests for the 3DES and AES algorithms. Other tests run at startup include a power on self-test run by the BIOS and a virus scan/partition scan run by the TOE. A continuous random number generator test is performed on the ANSI X9.17 PRNG. ***(FPT_AMT.1, FPT_TST.1)***

When communications have been lost with the TOE's central distribution server, the TOE will continue to enforce the various access control policies. The TOE has the capability to use redundant servers and will attempt to access another designated distribution server should communications with the default server fail. ***(FPT_FLS.1)*** The TOE protects TSF data being transferred between physically separated parts of the TOE from unauthorized disclosure and modification by encrypting the data. ***(FPT_ITT.1)***

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

6.1.5 Auditing

The TOE has the capability to capture an audit record for many different events. For each of the events the date/time, type of event, users 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 and shutdown events of the computer (auditing starts and stops with the startup and shutdown of the computer)
- All attempts by any user to authenticate to the TOE
- All attempts by users to access locked accounts
- All changes to a user's authentication data
- All attempts to uninstall the TOE from the system
- Account locked out due to exceeding the maximum number of unsuccessful logon attempts
- Changes to the system time
- All changes to the user database
- Use of the Remote Help feature

The TOE provides the Log Viewer utility for the authorized administrator to view the TOE's audit logs. Access to this utility is restricted to authorized administrators of the TOE and the logs are stored in an encrypted form. This combination of access control and encryption is used to prevent modification and deletion of the locally stored audit logs. The Log Viewer decrypts and 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. Authorized users are also able to review logs, but only those on the local computer. ***(FAU_SAR.1, FAU_SAR.2, FAU_SAR.3, FAU_STG.1)***

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

6.1.6 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 an earlier version of the product that has been certified as meeting the requirements of FIPS 140-1 (product evaluation) and is therefore compliant with those requirements. The cryptographic algorithms are certified as compliant with FIPS 46-3 (3DES algorithm) and FIPS 197 (AES algorithm).

The TOE implements the following cryptographic key management functions:

- **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 because a new database is created, and this new database overwrites the old one that contained the removed user's partition key. Also, to zeroize all keys, the System Administrator may uninstall the product. ***(FCS_CKM.4)***

- **Archiving:** The TOE does support key archiving. The user database file can be copied to a remote directory for recovery purposes. The file is encrypted with AES, and the partition keys in the user database are encrypted with AES. The TOE also implements recovery keys (K_R). A key recovery file is created for backup purposes if a partition is damaged. This key recovery file contains an additional copy of K_P for use in emergency recovery. The file is AES encrypted using a key recovery key (K_R). For each pair of users designated as a pair of key recovery operators, a copy of K_R is encrypted with K_{U1} and K_{U2} using the 2x5 AES matrix. ***(FCS_CKM.3)***

- **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 (key size is dependent on the algorithm and license) is internal to the TOE and is generated by a FIPS 140-1 compliant Key Generation algorithm. The recovery key is encrypted with the user ID and password hash of the associated users. 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. All archive and recovery keys are also encrypted before being stored. ***(FCS_CKM.1)***

The TOE performs the following cryptographic operations:

- **Key encryption and decryption:** All secret and private keys are encrypted and stored in the user database. The user selects a username and password at installation time. The password is then securely hashed using a one-way process that includes an array of 10 times AES (key size is dependent on the algorithm and license) used in an irreversible way, and the result of the hash is the key used to encrypt the partition key. The hash itself is not stored on the disk. Each successful logon with correct username and password recreates the hash to decrypt the key for the partition. The keys required to decrypt user data are not stored in the clear. Correct authentication of a user will generate the unique hash necessary to decrypt the user's partition key. ***(FCS_COP.1(b))***

- **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 first encrypted, and the result 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))***

Security Functional Requirements: FCS_CKM.1, FCS_CKM.3, FCS_CKM.4, FCS_COP.1(a), FCS_COP.1(b).

6.1.7 Fault tolerance

When a TOE-protected system loses contact with the Pointsec Distribution Server, the TOE provides the administrator with the capability to identify an additional three Pointsec Distribution Servers for redundancy. As a result, if the default Distribution Server is offline, or the system is unable to contact the Pointsec Distribution Server, the system will attempt to communicate with one of the other identified Distribution Servers. While a protected system is unable to contact a distribution 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. ***(FRU_FLT.1.)***

Security Functional Requirement: FRU_FLT.1.

6.1.8 Trusted path

For initial interactive 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 using the Ctrl-Alt-Del key sequence or a system reset, which is always captured by the TSF (i.e., it cannot be intercepted by an untrusted process), and 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). ***(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 assurance requirements:

- Process Assurance;
- Delivery and Guidance;
- Design Documentation;
- Tests; and,
- Vulnerability Assessment.

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 4.3 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 4.3 Life Cycle Management Plan.

Assurance Requirements: ALC_DVS.1, ALC_LCD.1, ALC_TAT.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 4.3 into the evaluated configuration. These procedures are documented in the Pointsec PC 4.3 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 4.3 Administrator Guide. User guidance is documented in the Pointsec PC 4.3 User Guide CBT.

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 4.3 to describe all aspects of the TOE security design, architecture, mechanisms, and interfaces. The documents that comprise this set are as follows:

- Pointsec PC 4.3 Functional Specification: A document that defines the interfaces and functionality of the TOE.
- Pointsec PC 4.3 High Level Design: A high-level architectural description of the TOE that defines the system through a set of subsystems.
- Pointsec PC 4.3 Low Level Design: A more detailed representation of the TOE that refines subsystems into modules.
- Pointsec PC 4.3 Implementation Representation: A representation of the source code used to implement the TOE.
- Pointsec PC 4.3 Security Policy Model: The Security Policy document fully presents an informal security model for the TOE.
- Pointsec PC 4.3 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, specifically describing all test cases and variations necessary to demonstrate that all security checks and effects related to the API are correctly implemented. 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 4.3 Administrator Guide. The Pointsec PC 4.3 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 4.3 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.ACCESS_CONTROL	T.ACCESS T.TSF_DATA
O.MEDIA_ACCESS	T.SUBVERT T.REMOVE_DISK
O.MANAGE	P.MANAGE

TOE Security Objectives	Threats and Organizational Policies
O.AUDIT	T.AUDIT_CORRUPT P.ACCOUNTABILITY 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.RESIDUAL_INFO	T.OBJECT_REUSE
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.ACCESS_CONTROL – This objective counters the threats T.ACCESS and T.TSF_DATA by requiring each user be authenticated before any access to the TOE and its protected resources is granted. 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 internal TOE communications are protected, countering the threat T.TRANSIT.

O.CRYPTO_KEYS – This objective implements the security policy P.CRYPTO_KEYS which ensures that all cryptographic key operations (generation, protection, destruction and protection) are performed in compliance to FIPS 140-1 (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 140-1 (product evaluation), FIPS 46-3 (3DES) and FIPS 197 (AES) specifications as appropriate.

O.RESIDUAL_INFO – This objective counters the threat T.OBJECT_REUSE by ensuring that previous authentication data can not be reused for unauthorized access to the TOE.

O.FAULT_TOLERANCE – This objective enforces the security policy P.FAULT_TOLERANCE by ensuring that the access control functions of the TOE will continue to operate if communications with the central administration servers are 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.

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

TOE Security Objectives for the IT Environment	Assumptions
OE.TIME_SOURCE	A.TIME
OE.SERVER	A.SERVER

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 server to be used as a distribution point for intra-TOE communications, log, recovery, update and installation files.

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 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.ACCESS_CONTROL	O.MEDIA_ACCESS	O.MANAGE	O.AUDIT	O.PROTECT	O.TRUSTED_PATH	O.DATA_TRANFER	O.CRYPTO_KEYS	O.CRYPTO_OPS	O.RESIDUAL_INFO	O.FAULT_TOLERANCE
FAU_GEN.1					X							
FAU_GEN.2					X							
FAU_SAR.1					X							
FAU_SAR.2					X							

SECURITY FUNCTIONAL REQUIREMENT	O.AUTHORIZATION	O.ACCESS_CONTROL	O.MEDIA_ACCESS	O.MANAGE	O.AUDIT	O.PROTECT	O.TRUSTED_PATH	O.DATA_TRANFER	O.CRYPTO_KEYS	O.CRYPTO_OPS	O.RESIDUAL_INFO	O.FAULT_TOLERANCE
FAU_SAR.3					X							
FAU_STG.1					X							
FCS_CKM.1									X	X		
FCS_CKM.3									X	X		
FCS_CKM.4									X	X		
FCS_COP.1(a)			X							X		
FCS_COP.1(b)									X	X		
FDP_ACC.2(a)		X										
FDP_ACC.2(b)		X										
FDP_ACF.1(a)		X										
FDP_ACF.1(b)		X										
FDP_RIP.1											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(a)				X								

SECURITY FUNCTIONAL REQUIREMENT	O.AUTHORIZATION	O.ACCESS_CONTROL	O.MEDIA_ACCESS	O.MANAGE	O.AUDIT	O.PROTECT	O.TRUSTED_PATH	O.DATA_TRANFER	O.CRYPTO_KEYS	O.CRYPTO_OPS	O.RESIDUAL_INFO	O.FAULT_TOLERANCE
FMT_MOF.1(b)				X								
FMT_MOF.1(c)				X								
FMT_MSA.1				X								
FMT_MSA.2				X								
FMT_MSA.3				X								
FMT_MTD.1(a)				X								
FMT_MTD.1(b)				X								
FMT_MTD.1(c)				X								
FMT_MTD.1(d)				X								
FMT_MTD.1(e)				X								
FMT_MTD.1(f)				X								
FMT_MTD.2				X								
FMT_REV.1				X								
FMT_SAE.1				X								
FMT_SMF.1				X								
FMT_SMR.1				X								
FPT_AMT.1						X						
FPT_FLS.1						X						
FPT_ITT.1								X				
FPT_RVM.1						X						
FPT_SEP.1						X						

SECURITY FUNCTIONAL REQUIREMENT	O.AUTHORIZATION	O.ACCESS_CONTROL	O.MEDIA_ACCESS	O.MANAGE	O.AUDIT	O.PROTECT	O.TRUSTED_PATH	O.DATA_TRANFER	O.CRYPTO_KEYS	O.CRYPTO_OPS	O.RESIDUAL_INFO	O.FAULT_TOLERANCE
FPT_TST.1						X						
FRU_FLT.1												X
FTP_TRP.1							X					

In addition to the above table, the environmental requirement, SFR FPT_STM.1 maps to the objective OE.TIME_SOURCE.

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.

FIA_ATD.1 and FIA_UAU.5 require that each user be uniquely identified by one of five authentication mechanisms, fixed passwords, dynamic/challenge-response tokens, smart cards, Remote Help, or Windows Password Change. 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_ALF.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.ACCESS_CONTROL

FDP_ACC.2(b) requires the TOE prevent unauthorized access to the System Area. Only authorized users are allowed to access the System Area.

FDP_ACF.1(b) requires that the TOE enforce security roles with unique privileges related to System Area access by authenticated users.

FDP_ACC.2(a) and FDP_ACF.1(a) require the TOE prevent unauthorized access to the encrypted partitions protected by the TOE by requiring successful user authentication to access the encryption keys.

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_MTF.1(b) 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(a) and FMT_MTD.1(a) provide authorized administrators with the ability to set the location where audit records will be stored as well as which security roles can query, delete and export the audit trail.

FMT_MSA.1 places restrictions on access to the System Area. These controls include creating new accounts, modifying security privileges/roles, authentication data and partitions keys, and they must be limited to authorized administrators only.

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

FMT_MTD.1(d) 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(e), FMT_MTD.1(f), 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.

FMT_MOF.1(c) and FMT_MTD.1(c) restrict the ability to change the cryptographic characteristics of the TOE to only authorized administrators, including creating and clearing partition keys.

FMT_MSA.2 requires the TOE to use only secure values for cryptographic operations such as when generating encryption keys.

FMT_MSA.3 ensures the TOE has a restrictive default setting for new user accounts.

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_AMT.1 and 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 communications failure with the central server 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.1 ensures 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 the TOE will protect internal data as it is transferred between different components of the TOE.

O.CRYPTO_KEYS

FCS_CKM.1, FCS_CKM.3 and FCS_CKM.4 require all cryptographic keys to be generated, protected, archived, used and deleted in accordance with FIPS 140-1 (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.

O.CRYPTO_OPS

FCS_COP.1(a), FCS_COP.1(b), FCS_CKM.1, FCS_CKM.3 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 140-1 (product evaluation), FIPS 46-3 (3DES) and FIPS 197 (AES) specifications as appropriate.

O.RESIDUAL_INFO

FDP_RIP.1 requires that previous authentication (such as old passwords) be unavailable for use once modified by an authorized user.

O.FAULT_TOLERANCE

FRU_FLT.1 defines that the TOE will continue to enforce all security policies in the event of a communications failure with the central administration server.

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.

8.2.2 Security Assurance Requirements Rationale

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

Table 8: SFRs and associated dependencies

Requirement No.	Functional Requirements	Dependencies	Dependency	Dependency Met
1	FAU_GEN.1	FPT_STM.1	36	✓
2	FAU_GEN.2	FAU_GEN.1	1	✓
		FIA_UID.1	21 (UID.2)	✓
3	FAU_SAR.1	FAU_GEN.1	1	✓
4	FAU_SAR.2	FAU_SAR.1	3	✓
5	FAU_SAR.3	FAU_SAR.1	3	✓
6	FAU_STG.1	FAU_GEN.1	1	✓
7	FCS_CKM.1	FCS_CKM.2 or FCS_COP.1	10	✓ (FCS_COP.1)
		FCS_CKM.4	9	✓
		FMT_MSA.2	24	✓
8	FCS_CKM.3	FDP_ITC.1 or FCS_CKM.1	7	✓ (FCS_CKM.1)
		FCS_CKM.4	9	✓
		FMT_MSA.2	24	✓
9	FCS_CKM.4	FDP_ITC.1 or FCS_CKM.1	7	✓ (FCS_CKM.1)
		FMT_MSA.2	24	✓
10	FCS_COP.1	FDP_ITC.1 or FCS_CKM.1	7	✓ (FCS_CKM.1)
		FCS_CKM.4	9	✓
		FMT_MSA.2	24	✓
11	FDP_ACC.2	FDP_ACF.1	12	✓
12	FDP_ACF.1	FDP_ACC.1	11 (ACC.2)	✓
		FMT_MSA.3	25	✓
13	FDP_RIP.1	No dependencies	-	✓
14	FIA_AFL.1	FIA_UAU.1	17 (UAU.2)	✓
15	FIA_ATD.1	No dependencies	-	✓
16	FIA_SOS.1	No dependencies	-	✓
17	FIA_UAU.2	FIA_UID.1	21 (UID.2)	✓
18	FIA_UAU.4	No dependencies	-	✓

Requirement No.	Functional Requirements	Dependencies	Dependency	Dependency Met
19	FIA_UAU.5	No dependencies	-	
20	FIA_UAU.7	FIA_UAU.1	17 (UAU.2)	
21	FIA_UID.2	No dependencies	-	
22	FMT_MOF.1	FMT_SMF.1	29	
		FMT_SMR.1	30	
23	FMT_MSA.1	FDP_ACC.1 or FDP_IFC.1	11 (ACC.2)	(FDP_ACC.2)
		FMT_SMF.1	29	
		FMT_SMR.1	30	
24	FMT_MSA.2	ADV_SPM.1	CC EAL4	
		FDP_ACC.1 or FDP_IFC.1	11 (ACC.2)	(FDP_ACC.2)
		FMT_MSA.1	23	
		FMT_SMR.1	30	
25	FMT_MSA.3	FMT_MSA.1	23	
		FMT_SMR.1	30	
26	FMT_MTD.1	FMT_SMF.1	29	
		FMT_SMR.1	30	
27	FMT_REV.1	FMT_SMR.1	30	
28	FMT_SAE.1	FMT_SMR.1	30	
		FPT_STM.1	36	
29	FMT_SMF.1	No dependencies	-	
30	FMT_SMR.1	FIA_UID.1	21 (UID.2)	
31	FPT_AMT.1	No dependencies	-	
32	FPT_FLS.1	ADV_SPM.1	CC EAL4	
33	FPT_ITT.1	No dependencies	-	
34	FPT_RVM.1	No dependencies	-	
35	FPT_SEP.1	No dependencies	-	
36	FPT_STM.1	No dependencies	IT Environment	
37	FPT_TST.1	FPT_AMT.1	31	
38	FRU_FLT.1	FPT_FLS.1	32	
39	FTP_TRP.1	No dependencies	-	
40	FMT_MTD.2	FMT_MTD.1	26	
		FMT_SMR.1	30	

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 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 and FIA_UAU.2 of guessing a fixed password is consistent with the security objectives of the TOE. 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 two authentication mechanisms: dynamic/challenge-response tokens and smart cards. Both of these authentication mechanisms are two-factor, employing separate, physical devices to provide the appropriate authentication to the TOE, and this identity information is cryptographic in nature.

The dynamic/challenge-response token uses an internal DES key to generate challenge and response sequences. Each different challenge will generate a different response, and requires physical control of the token to generate the response from the TOE's challenge. To generate the correct response, the user must also have entered the correct four digit PIN into the token. An incorrect PIN will still generate a response, but it will be invalid and not allow authentication to the TOE.

The smart card stored a private key internally, which is used to provide identification for the user. This key is queried via cryptographic functions to provide authentication data that can be verified. The smart card also requires a PIN to gain access to internal data, including the cryptographic identity. The smart card is locked after three unsuccessful PIN attempts, which then requires administrative access to unlock. This method of authentication requires physical control of the smart card and physical access to the TOE, since the identity is read directly into the TOE; the user is only required to enter the correct PIN to unlock the smart card functionality. Without the correct combination of PIN and smart card, access to the TOE will be denied.

The claimed SOF-medium also applies to the Remote Help authentication mechanisms. These mechanisms require independent identification over the phone (assumption A.PHONE_DATA). Several recommended

approaches to verification are given in the Admin Guidance. The actual mechanism itself is protected through the use of strong, one-time use keys, generated by the ANSI X9.17 PRNG. Since these keys are randomly generated for each session using an accepted RNG, they provide security assurances equal to at least SOF-medium.

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.

Table 9: Mapping of SFRs to Security Functions

REQUIREMENT	ACCESS CONTROL	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.3						X		
FCS_CKM.4						X		
FCS_COP.1(a)						X		

REQUIREMENT	ACCESS CONTROL	IDENTIFICATION AND AUTHENTICATION	SECURITY MANAGEMENT	SELF-PROTECTION	AUDITING	CRYPTOGRAPHIC SUPPORT	FAULT TOLERANCE	TRUSTED PATH
FCS_COP.1(b)						X		
FDP_ACC.2(a)	X							
FDP_ACC.2(b)	X							
FDP_ACF.1(a)	X							
FDP_ACF.1(b)	X							
FDP_RIP.1	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(a)			X					
FMT_MOF.1(b)			X					
FMT_MOF.1(c)			X					
FMT_MSA.1			X					
FMT_MSA.2			X					
FMT_MSA.3			X					
FMT_MTD.1(a)			X					
FMT_MTD.1(b)			X					

REQUIREMENT	ACCESS CONTROL	IDENTIFICATION AND AUTHENTICATION	SECURITY MANAGEMENT	SELF-PROTECTION	AUDITING	CRYPTOGRAPHIC SUPPORT	FAULT TOLERANCE	TRUSTED PATH
FMT_MTD.1(c)			X					
FMT_MTD.1(d)			X					
FMT_MTD.1(e)			X					
FMT_MTD.1(f)			X					
FMT_MTD.2			X					
FMT_REV.1			X					
FMT_SAE.1			X					
FMT_SMF.1			X					
FMT_SMR.1			X					
FPT_AMT.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	
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				
ACM_CAP.4	X				

SARs	Process Assurance	Delivery and guidance	Design Documents	Test	Vulnerability Assessment
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_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