PGP® Desktop: Enterprise Whole Disk Encryption Only Edition

Version 9.10.0

Security Target

EAL 4 augmented ALC_FLR.1

Release Date:	March 30, 2010
Document ID:	08-1622-R-0004
Version:	1.0

Prepared By: M. McAlister
InfoGard Laboratories

Prepared For: PGP®

200 Jefferson Drive
Menlo Park, CA 94025 USA

Table of Contents

List of Tables

List of Figures

1 Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), conformance claims, ST organization, document conventions, and terminology. It also includes an overview of the evaluated product.

1.1 Organization

- **Security Target Introduction (Section 1)** – Provides identification of the TOE and ST, an overview of the TOE, an overview of the content of the ST, document conventions, and relevant terminology. The introduction also provides a description of the TOE security functions as well as the physical and logical boundaries for the TOE, the hardware and software that make up the TOE, and the physical and logical boundaries of the TOE.
- **Conformance Claims (Section 2)** – Provides applicable Common Criteria (CC) conformance claims, Product Profile (PP) conformance claims and Assurance Package conformance claims.
- **Security Problem Definition (Section 3)** – Describes the threats, organizational security policies, and assumptions pertaining to the TOE and the TOE environment.
- **Security Objectives (Section 4)** – Identifies the security objectives for the TOE and its supporting environment as well as a rationale that TOE objectives are sufficient to counter TOE threats identified for the TOE. This section also provides a justification for each assumption and the security objectives for the environment which cover that assumption.
- **Extended Components Definition (Section 5)** – Presents components needed for the ST but not present in Part II or Part III of the Common Criteria Standard.
- **Security Requirements (Section 6)** – Presents the Security Functional Requirements (SFRs) met by the TOE and the security functional requirements rationale. Provides pointers to all other rationale sections, to include the rationale for the selection of IT security objectives, requirements, and the TOE summary specifications as to their consistency, completeness, and suitability
- **Summary Specification (Section 7)** – Describes the security functions provided by the TOE that satisfy the security functional requirements, provides the rationale for the security functions. It also describes the security assurance measures for the TOE as well as the rationales for the assurance measures.

1.2 Document Conventions

The CC defines four operations on security functional and assurance requirements. The conventions below define the conventions used in this ST to identify these operations. When

NIAP interpretations are included in requirements, the changes from the interpretations are displayed as refinements.

Assignment: **indicated with bold text**

<u>Selection:</u> <u>indicated with underlined text</u>

Refinement: ***additions indicated with bold text and italics***

deletions indicated with strike-through ***~~bold text and italics~~***

Iteration: indicated with typical CC requirement naming followed by a lower case letter for each iteration (e.g., FMT_MSA.1a)

The explicitly stated requirements claimed in this ST are denoted by the "_EXP" extension in the unique short name for the explicit security requirement.

1.3 Document Terminology

Please refer to CC v3.1 Part 1 Section 4 for definitions of commonly used CC terms.

1.3.1 ST Specific Terminology

Authorized User — Refers to the local user of the TOE application. Since authentication during bootup of the platform is required in order to access/decrypt the protected physical disk, this user is referred to as an authorized user.

Disk Access Key — Refers to a symmetric key created from the user passphrase/token used to encrypt the Link Key. The Disk Access Key is unique to the disk group.

Link Key — Refers to a symmetric key used to link multiple disks within a single platform. Note role in decrypt sequence under Disk Access Key definition.

Managed WDE — Refers to the TOE application, which is the PGP Desktop platform with only WDE enabled. The managed prefix indicates that it includes the ability to be remotely managed through a PGP Universal Server in the Operational Environment.

Partition Encryption — Refers to the encryption of a specific partition of the physical drive on the local platform by the TOE to one or more passphrases or cryptographic keys.

PGP Desktop — Refers to the portion of the PGP TOE software that executes as an application when selected following the Operating System boot process. The PGP Desktop is also a generic term used for the PGP application upon which WDE executes.

PGP Universal Server

Refers to a server component within the Operational Environment which can be used to manage multiple instances of PGP TOE installations throughout the deployed network. Within the context of this ST this could be either the PGP Universal Server or the Generic Equivalent as specified in Section 1.9.2.

Pre-Boot

Refers to security functions or actions taken prior to the platform Operating System completes the booting process. The authentication process for the TOE occurs during this stage.

Post-Boot

Refers to security functions or actions taken following the platform Operating System completing the booting process.

Session Key

Within this ST, this refers to the symmetric key used by the TOE for whole disk or partition encryption. This key is encrypted by the Link Key.

Virtual Disk Encryption

Refer to the creation of a separate disk representation on the physical drive on the local platform of a specified size which is encrypted by the TOE to one or more passphrases or cryptographic keys. (excluded from TOE, CC evaluated configuration)

WDRT "single use"

Refers to a TOE feature where the recovery token (WDRT) can only be used a single time and a new token is generated by the TOE and is sent to the Universal Server in the Operational Environment. "Single use" refers to a single usage to unlock the disk with a complete, uninterrupted boot process and "Normal Mode" Operating System logon. In the event of an interrupted boot or safe mode boot, the token does not generate to assure a valid WDRT is always available for the platform.

Windows

Refers to the Operating System component of the TOE: Microsoft Windows XP Professional SP3

Whole Disk Encryption

Refers to the encryption of the entire physical drive on the local platform to one or more passphrases or cryptographic keys by the TOE , requiring passphrase authentication at the Disk BIOS level to complete the boot process.

Whole Disk Recovery Token	Refers to a passphrase/token created by the mWDE TSF during whole disk/partition encryption operations. This produces an additional passphrase that can be used to access the physical drive or partition in the event the authorized user forgets the user created passphrase. This recovery passphrase/token is stored on the PGP Universal Server.

1.3.2 Acronyms

3DES	Triple DES
AES	Advanced Encryption Standard
CC	Common Criteria
DES	Data Encryption Standard
DLL	Dynamic Link Library
FIPS	Federal Information Processing Standard
GUI	Graphic User Interface
mWDE	Managed WDE
NIST	National Institute (of) Standards & Technology
NTFS	NT File System (Windows File System)
OEAP	Optimal Asymmetric Encryption Padding
S2K	String to Key operation (creates cryptography key from passphrase string)
PGP	Pretty Good Privacy (and sponsor corporate name)
PIN	Personal Identification Number
RNG	Random Number Generator
RTF	Rich Text Format
SDK	Software Developers Kit
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSC	TOE Scope of Control
TSF	TOE Security Functionality
UI	User Interface (subsystem)
WDRT	Whole Disk Recovery Token
XML	Extended Markup Language

1.4 Identification

TOE Reference:	PGP® Desktop: Enterprise Whole Disk Encryption Only Edition, Version 9.10.0, Build 596.
ST Reference:	PGP® Desktop: Enterprise Whole Disk Encryption Only Edition Version 9.10.0 Security Target EAL 4 augmented ALC_FLR.1
ST Version:	1.0
ST Publish Date:	March 30, 2010
ST Author:	Mike McAlister (InfoGard)
PP Identification:	Not Applicable

1.5 Overview

The PGP managed whole disk encryption (mWDE) application allows the local user to encrypt the entire physical hard disk or selected partitions on the installed platform. Upon activation of the Whole Disk Encryption function, the complete contents of the physical drive are encrypted using FIPS 140-2 validated symmetric cryptography, protecting the operating system and data stored within the physical drive from unauthorized access when shutdown. In the event the computer is stolen or even if the drive is removed from the platform, the data remains safely encrypted. Partition Encryption protects the physical drive data at the partition level in the same manner. Using either of these options requires authentication during the platform boot process in order to access the physical disk. Once the encryption step is complete, data is never written to the encrypted disk or partition in an unencrypted form.

The TOE supports encryption of the entire physical drive or only specified partition(s).

Upon boot of the protected drive, the user is presented a dialog for the entry of user credentials required to access drive contents and decrypt content. The mWDE TOE includes a disk BIOS component that replaces the existing Master Boot Record (MBR) and provides a protected application interface for authentication. This pre-boot authentication assures that the user is successfully authenticated prior to being allowed access to any drive resources. In addition, the TOE includes a two factor authentication option which supports the use of hardware tokens/smartcards or the use of a Trusted Platform Module (TPM) for authentication.

The TOE also supports Windows single-sign-on based on the use of the Windows account logon credentials for Whole Disk/Partition encryption operations allowing the TOE to both unlock the

WDE/Partition encryption during boot and log onto the applicable Windows account using existing domain password rules.

The mWDE application includes the PGP Desktop application infrastructure which provides a full function Graphic User Interface (GUI) supporting configuration and management of the mWDE application.

The “managed” aspect relates to the management of the deployed TOE application using a PGP Universal Server which allows server administrators to manage multiple mWDE installations on deployed mWDE platforms throughout the organization. The PGP Universal Server is part of the Operational Environment and is not included with the TOE. The mWDE application simply provides the ability to interface with this server. This server can alternatively be constructed using Open Source components that support the XML/SOAP based commands necessary to support the mWDE managed functions or through a PGP Universal Server that may pre-exist in the deployment environment.

The PGP Universal Server, as part of the Operational Environment, allows PGP Universal Server Administrators to enforce mWDE related security policies which specify what mWDE operations may be implemented locally. This includes the ability for local users to encrypt/decrypt local drives, allowing the use of recovery tokens and enabling/disabling PGP Desktop features. The mWDE TOE provides for a recovery key to be automatically created locally during encryption processes and stored on the PGP Universal Server platform in the Operational Environment. This allows PGP Universal Server Administrators to provide the recovery key (one-time use passphrase) to local users in the event the user created passphrase is forgotten. Once the recovery passphrase is used and the platform completes boot and a “Normal Mode” OS login process, the mWDE application initiates immediate generation of a new recovery key which is then forwarded to the PGP Universal Server. This server also provides a centralized logging repository for deployed mWDE applications throughout the network.

1.5.1 TOE Type

The TOE is classified as a **Sensitive Data Protection** application for Common Criteria purposes. The TOE is made up of software components.

1.5.2 Non-TOE Hardware/Software required by the TOE

Component	Description
Hardware platform for TOE installation	General purpose laptop or desktop capable of running Microsoft® Windows XP Professional SP3

Required Monitor & Peripherals based on platform	USB keyboard, USB mouse and video monitor based on hardware options selected above – desktop will require peripheral devices.
Trusted Platform Module as part of hardware listed above when applicable	As noted in Table 4: Supported Trusted Platform Modules for TOE below
PGP Universal Server hardware	Hardware for PGP Universal Server General Purpose Server capable of running XML/SOAP commands to/from TOE installations For PGP Universal Server option hardware requirements see http://www.pgp.com/products/universal_server/tech_specs.html
Smartcard/USB Token & associated reader when applicable	As noted in Table 3: Supported Smartcards/USB tokens for TOE below.
OpenLDAP authentication server platform	Hardware for OpenLDAP server or equivalent: General Purpose server capable of hosting OpenLDAP 2.4.6 or greater see: http://www.openldap.org/

Table 1: Non-TOE Hardware Components

Environment	Component	Description
Environment	Underlying Operating System for TOE application	Microsoft Windows XP Professional SP3
Environment	PGP Universal Server 2.10.0 software *or* Generic implementation: OS support capable of running a Web Server to host XML/SOAP - gSOAP toolkit 2.7.10	PGP Universal Server software component PGP Universal Server or Generic Open Source option
Environment	Trusted Platform Module software	Per Table 4: Supported Trusted Platform Modules for TOE
Environment	Smartcard/USB Token software when applicable	Per Table 3: Supported Smartcards/USB tokens for TOE
Environment	OpenLDAP authentication server software or LDAP equivalent	OpenLDAP version 2.4.6 or greater or equivalent LDAP server see: http://www.openldap.org/

Table 2: Non-TOE Software Components

The following smartcards and/or tokens are supported by the TOE:

ActiveIdentity ActivClientCAC cards, 2005 models
Aladdin eToken 64K, 2048-bit RSA-capable1
Aladdin eToken PRO USB Key 32K, 2048-bit RSA-capable1
Aladdin eToken PRO without 2048-bit capability (older smart cards)1
Athena ASEKey Crypto USB Token for Microsoft ILM2
Athena ASECard Crypto Smart Card for Microsoft ILM2
EMC RSA SecurID SID800 Token3
Charismathics CryptoIdentity plug 'n' crypt Smart Card only stick
S-Trust StarCOS smart card4
Rainbow iKey 3000

Table 3: Supported Smartcards/USB tokens for TOE

The following Trusted Platform Modules (TPM) are supported by the TOE:

Hewlett-Packard Compaq nx6325 (Infineon TPM with HP BIOS)
Dell D620/D630 (Broadcom TPM)
Lenovo ThinkPad T60 (Atmel TPM)
Fujitsu LifeBook T2010, (Infineon TPM with Phoenix BIOS)
Panasonic Toughbook T5, W5, or Y5 (Infineon TPM with Matsushita BIOS)

Table 4: Supported Trusted Platform Modules for TOE

Since the CC Evaluated version of Whole Disk Encryption does not include key generation/management functions, key used in conjunction with the above supported Smartcards/Token must originate from outside the scope of this product. Keypairs used with supported Smartcards/Tokens must meet the following requirements:

1. Keypairs must be of type:
    a. RSA

b. PKCS12 certificate supporting (keyUsage) key and data encipherment, (encryption decryption)

2. Keys must conform to applicable Smartcard/Token compatibility requirements and be sized between 1024 and 2048 bits and are used for encryption/decryption.

1.6 Architecture Overview

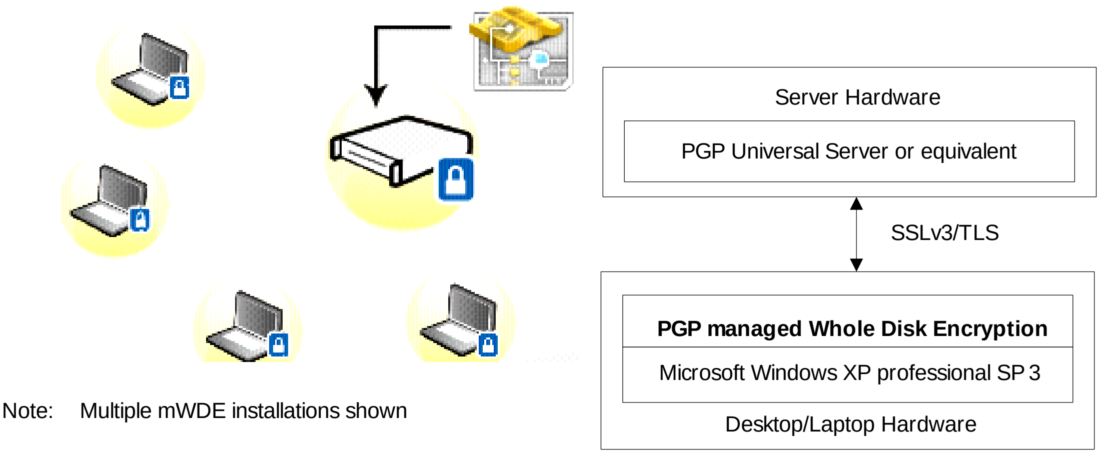
Server Hardware
PGP Universal Server or equivalent
SSLv3/TLS
PGP managed Whole Disk Encryption
Microsoft Windows XP professional SP 3
Desktop/Laptop Hardware
Note: Multiple mWDE installations shown

Figure 1: TOE network architecture

1.7 Architecture Description

The TOE software component is divided into two major sections: The PGP desktop and the mWDE disk BIOS section. The PGP desktop component is the main application program which includes the GUI management interface and engages following the boot process of the local machine. The disk BIOS portion of the TOE application engages during power-up and provides the protected (pre-boot) user interface used to enter the passphrase/access the token key, validate authentication credentials and initiate decryption of the TSF encrypted boot disk to allow for system startup. The software architecture is divided into two sections: 1.7.1 - The PGP Desktop component that refers to the application running post boot and 1.7.2 containing the BIOS portion of the TOE that executes during the platform startup/boot processes.

1.7.1 PGP managed Whole Disk Encryption software component: PGP Desktop

The PGP desktop software component includes the aspects of the application which operate following the startup process on the deployed platform. These subsystems provide the base software engine, GUI based user interface, cryptographic subsystems and interface to the underlying disk storage media.

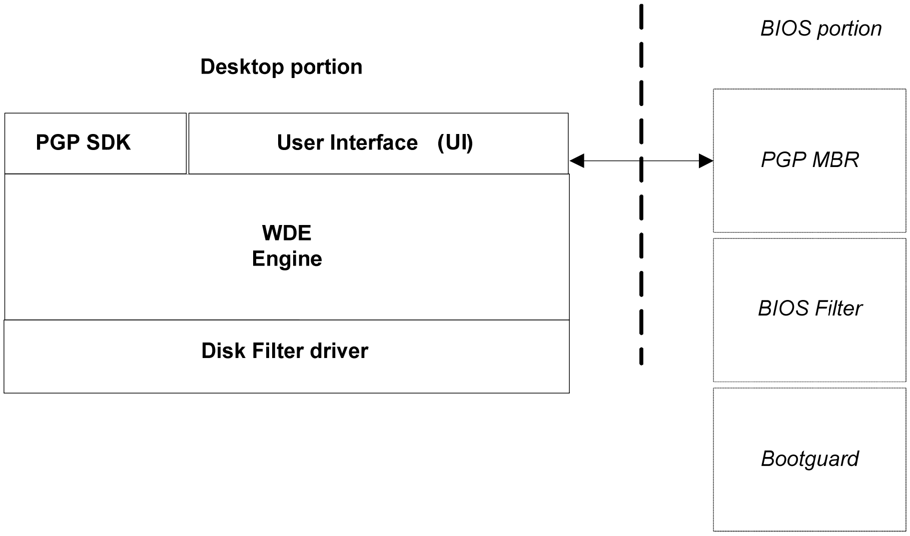

Figure 2: TOE Internal Architecture: PGP Desktop portion

* note the relative size and location of subsystems does not connote architectural relationships or interfaces – for illustration purposes of subsystem components only

1.7.1.1 WDE Engine Subsystem

The WDE Engine subsystem provides the core application engine upon which mWDE executes. This subsystem, written in "C^{++}", and implemented as a "Windows DLL module", acts as a central hub for operations performed by the TOE and acts upon User Requests to perform operations such as key generation, local log access, disk/partition encryption and user management activities such as key management. The WDE Engine subsystem communicates with the Disk Filter driver subsystem to execute the various application operations against the physical disk. The WDE Engine also includes the PGPTray component used to access log events and store them in the applicable application files for integration into the log file for the application.

1.7.1.2 User Interface (UI) Subsystem

The User Interface subsystem provides the Graphical User Interface (GUI) based management interface for the TOE. This subsystem is written in "C^{++}" and provides a series of user management screens that allow for local user management of keys, partition-based encryption and whole disk encryption activities. GUI elements for installed features are dynamically loaded based on registry settings that specify whether a feature is installed and enabled. Policy settings that may be passed from the PGP Universal Server are also enforced by the UI subsystem. In the event a selection is made from the GUI that is disallowed by policy, an error dialog is displayed that the feature is disabled for the authorized local user.

The User Interface also detects events related to user actions and generates log records for the application. The UI subsystem generates the log event, passes it to the Disk Filter Driver subsystem which signals the PGPTray application which stores the event in local application log files. These log records are saved to a local file and are also sent via SSLv3.1/TLSv1.0 to the PGP Universal Server in the Operational Environment.

1.7.1.3 PGP Software Developers Kit (SDK) Subsystem

The PGP software developer's kit cryptographic subsystem is a "FIPS 140-2 validated cryptographic module" in the form of a shared library binary and supports specific cryptographic functions provided by the TOE for the Desktop portion of the application software. This includes key generation, and cryptographic key destruction (zeroization). This subsystem also contains the desktop cryptographic libraries accessed to implement cryptographic functions by the application at the desktop (non-disk bios) level.

(FIPS 140-2 Cert. #1101)

1.7.1.4 Disk Filter driver Subsystem

The Disk Filter driver subsystem provides the driver support for the hard disk installed on the selected platform. This driver assists Operating System drivers accessing the disk by identifying which partitions/sectors are encrypted or are plaintext and works with the WDE engine and the PGP Cryptographic Engine to execute required encryption/decryption operations. The PGP Cryptographic Engine, a module within the Disk Filter driver subsystem, performs encryption/decryption operations (Post Boot) in conjunction with the Disk Filter driver, against data stored on the targeted disk once initial encryption is complete.

The mWDE uses this driver to access the PGP Cryptographic Engine and orchestrates access and perform encryption/decryption operations against the target disk in conjunction with the Operating System file system driver and disk device driver. This subsystem also identifies events requiring logging and calls the User Interface subsystem to create a log entry upon detection of an auditable event.

1.7.2 PGP managed Whole Disk Encryption software component: PGP® WDE Disk BIOS portion

The PGP mWDE disk bios portion of the application software construct contains the startup software that operates upon power up of the installed platform. This portion of the TOE application provides the PGP MBR, BIOS filter and Bootguard components which work in unison to provide the logon interface for passphrase/token key entry, authentication mechanisms and stored session key decryption used to authorize session startup and access the physical disk. These subsystems also provide the pre-boot operating environment which prevents unauthorized disk access prior to passphrase authentication. The pre-boot process executes exclusively in non-paged memory and only allows access to a 1 MB partition to the pre-boot program, for the purpose of storing the bootloader executable and caching logs created during pre-boot processes.

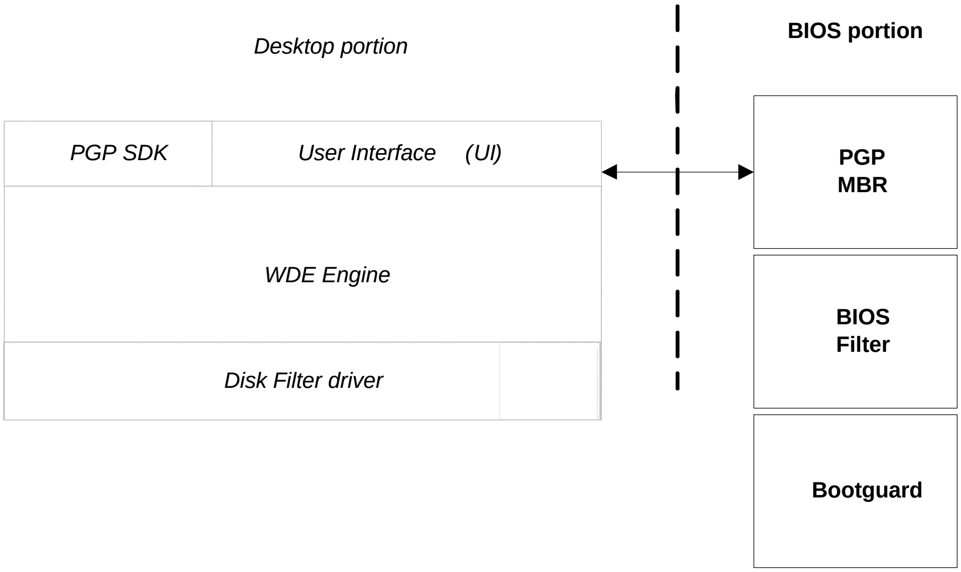

Figure 3: TOE Internal Architecture: PGP® WDE Disk BIOS portion

* note the relative size and location of subsystems does not connote architectural relationships or interfaces – for illustration purposes of subsystem components only

1.7.2.1 PGP Master Boot Record (MBR) Subsystem

The PGP Master Boot Record subsystem is a Master Boot Record designed to load the Bootguard executable, thereby temporarily preventing the system MBR from booting in the normal manner during startup. This assures that during startup, the PGP Bootloader (within the Bootguard subsystem) controls the startup process and successfully authenticates the user prior to allowing access to the disk BIOS or boot process.

1.7.2.2 BIOS Filter Subsystem

The BIOS Filter subsystem is loaded during the pre-boot process by the Bootguard executable. The BIOS filter provides a passive filter to control I/O during the PGP pre-boot process. The BIOS filter component includes a “C” based AES implementation for the purposes of decrypting disk information during the pre-boot process. While key material is provided by the Bootguard subsystem, the actual decryption activity is performed by the BIOS Filter subsystem during disk I/O operations. The BIOS filter also proxies disk I/O during the OS Bootstrap process during the PGP pre-boot stage of startup.

1.7.2.3 Bootguard Subsystem

The Bootguard subsystem represents the pre-boot executable (bootloader) that orchestrates the startup process to assure that no access is allowed to the encrypted drive, operating system or (non-PGP) disk BIOS prior to authentication. This subsystem includes the pre-boot application piece which presents the logon interface during system startup. Once the applicable passphrase (or token key) is entered by the user, the credentials are authenticated (hashed and used to decrypt the Disk Access key) and the BIOS filter is loaded to manage read/write operations. Bootguard then accesses the non-PGP disk partition and loads the OS (non-PGP) MBR which loads bootstrap code and device drivers and then accesses disk to commence with normal startup processes. The Bootguard also contains the drivers needed for two-factor authentication devices such as hardware tokens, smart cards etc.

This subsystem also transforms entered passphrases into cryptographic keys used to access the Disk Access, Link and Session Cryptographic keys. The Bootguard subsystem supports these functions during Pre-Boot operations.

This subsystem provides cryptographic support for hashing the entered passphrase/token key to decrypt and ultimately access the session key required for system startup. Algorithms used are Cryptographic Algorithm Validation Program (CAVP) validated.

1.8 TOE Description: Physical Boundaries

This section lists the TOE software product and illustrates the applicable boundaries of the product under evaluation..

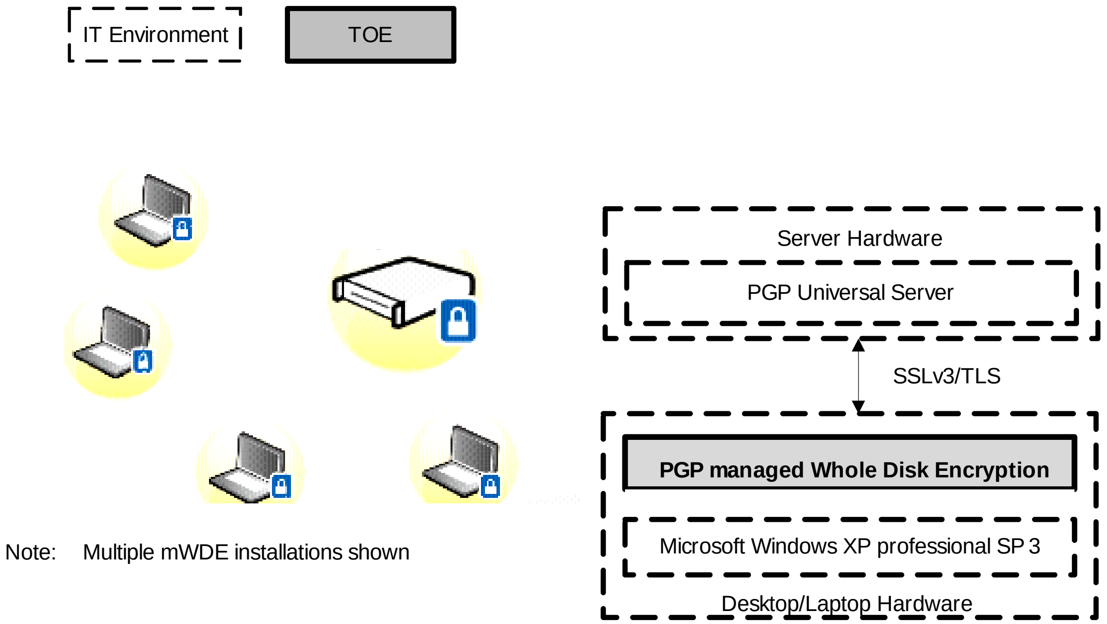
IT Environment
TOE
Server Hardware
PGP Universal Server
SSLv3/TLS
PGP managed Whole Disk Encryption
Microsoft Windows XP professional SP 3
Desktop/Laptop Hardware
Note: Multiple mWDE installations shown

Figure 4: TOE Physical Boundaries

1.8.1 Software Components

This table identifies the TOE software component.

TOE or Environment	Component	Description
TOE	PGP Whole Disk Encryption Version 9.10.0 build 596	TOE software

Table 5: Software Components

As noted in Section 1.5.2, the TOE is installed on a Microsoft Windows XP Professional SP3 Operating System environment housed in a General purpose laptop or desktop capable of running Microsoft® Windows XP.

1.8.2 Guidance Documents

The following guidance documents are provided with the TOE upon delivery in accordance with EAL 4 requirements:

a. PGP Whole Disk Encryption for Windows Quick Start Guide Version 9.10.0
b. PGP® Desktop for Windows User's Guide, PGP Desktop Version 9.10.0
c. PGP® Desktop Version 9.10 for Windows Release Notes (version 9.10.0)
d. Managed Whole Disk Encryption Common Criteria Supplemental, Version 9.10.0 Released March 2010

1.9 TOE Description: Logical Boundaries

This section contains the product features and denotes which are within the logical boundaries of the TOE. The following Security Functions are supported by the PGP® TOE:

- Security Audit
- Identification and Authentication
- Cryptographic Operations
- Security Management
- Protection of the TOE

1.9.1 Security Audit

The TOE application maintains a comprehensive logging capability that generates audit logs for selected security events relating to application usage and management.

Audit logs generated include three major categories of information local to the TOE application: Time/Date, Category (Info, Warning etc), and Message. The Time category specifies the date and time of the event and the Category specifies the type of event. The Category specifies a log type for sorting purposes. Under the Message category, the TOE includes the description of the event.

The User Interface subsystem within the Desktop portion of the mWDE software generates audit logs using the Bootguard subsystem for events related to the pre-boot process and by the UI subsystem for user initiated events after the machine is booted and the PGP TOE application is in use. For logging during pre-boot, the event is detected and generated by Bootguard which passes the information to the Disk Filter driver. The data is queued until the boot process completes and it is then written to the file system for local audit logs and is retrieved by the "ClientLib module" (WDE Engine component) which transfers the logged event in XML format to the PGP Universal Server in the Operational Environment.

Following the boot process the Disk Filter driver identifies disk related events and triggers the User Interface subsystem to log the event. The UI subsystem generates the event and passes it to the Disk Filter Driver subsystem, which signals the PGPtray application and the event is stored in local application log and transfers the log event to the PGP Universal Server. The managed aspect of the TOE also allows that audit logs may be centralized on the PGP Universal Server in the Operational Environment. Logs are sent from the User Interface subsystem to the Universal Server by the "ClientLib module" (part of the WDE Engine subsystem) using the Simple Object Access Protocol (SOAP) implemented via SSLv3.1/TLSv1.0.

Log content is identical between the log files saved locally and those that are passed to the PGP Universal Server in the Operational Environment except for the fact that disabling local logging is captured and passed to this server but is not stored in the local application logs. Within the TOE application local logging can be disabled, however, this has no effect on logs being generated and passed to the PGP Universal Server. Disabling logging within the application only results in those logs not being stored locally.

Audit logs for the TOE application are stored within the NTFS application storage folder (program files) resource in binary form. Any Post-Boot user may access audit records.

1.9.2 Identification and Authentication

The TOE application performs identification and authentication once the whole disk/partition encryption feature has been activated during the platform startup process and for selected TOE security management operations. With the whole disk encryption feature activated, the local user must be successfully identified and authenticated by the TOE through passphrase/smartcard or token key entry.

The TOE application supports the following options for authentication during boot:

- Passphrase and Single Sign-On Authentication
- Smartcard or Token-Based Authentication
- Trusted Platform Module (TPM) Authentication

The Passphrase option allows the user to specify a unique passphrase for the purpose of accessing the drive during PGP BIOS startup. The Passphrase is hashed using the PGP SDK cryptographic subsystem and is used to encrypt the Disk Access key which is used as a password to encrypt the Link Key which in turn is used to encrypt the Session Key, required to unlock the protected disk.. Once the correct passphrase is entered during the boot process, the Disk Access key is decrypted and used to decrypt the Link Key which in turn decrypts the Session Key securing the user data on the disk. In addition, the user can configure the TOE application to synchronize the PGP passphrase with the Windows logon so only the passphrase is required to unlock the physical drive and authenticate to the applicable Windows account.

The Smartcard or Token based authentication method supports the use of a PGP keypair stored on a hardware token or smart card for providing the authentication credentials during the PGP Bootguard (pre-boot) process. The Disk Access Key, which is unique to the disk group, is wrapped with a public key provided by the Token and is stored on the Token for use in TOE authentication. Drivers installed within the Bootguard are required to interact with the specified token or smartcard type. Hardware token and Smartcards supported by the TOE are listed in Table 3: Supported Smartcards/USB tokens for TOE.

Trusted Platform Module (TPM) authentication used by the TOE application for authentication to an mWDE protected disk resource relies on hardware on specified platforms that includes a TPM that utilizes 2048 bit RSA keys and brute force protection features. With this technique the disk is locked to the specific system through the use of the Storage Root Key (SRK) within the TPM. TPM version 1.1 and 1.2 are supported by the TOE for authentication support.

The machines firmware executes the PGP MBR which loads the Bootguard executable. The Bootguard loads the BIOS filter to manage pre-boot I/O and presents the login dialog to the user. The user enters the applicable passphrase or presents the token/smartcard; the credentials are hashed by the Bootguard subsystem to form a key which is used to decrypt a Disk Access key stored within the Bootguard keystore. Once the Disk Access key is released, the Link Key and Session Key are release in succession resulting in the Session Key being cached, allowing the boot process to proceed. In the event a token is in use for authentication, the entered passphrase represents a PIN number for access to the private RSA key used to decrypt the Disk Access key and ultimately decrypt the Session key stored within the Bootguard subsystem.

If the validation succeeds, the associated key is accessed from the Bootguard and is read into cache. Bootguard then allows disk access (beyond the limited PGP partition) and proceeds to load the platform MBR and execute the boot process.

TOE protection mechanisms are intended to protect data stored on hard disk resources through encryption and access control during startup boot up sequences only. As such, once the user has successfully authenticated during startup and consequently enabled access to the underlying Operating System, platform resources are accessible. A limited set of management functions require the local user to enter credentials to release required cryptographic keys post boot; however, no further protections against unauthorized access to user data are claimed by the TOE once the platform has been unlocked during start up boot processes.

1.9.3 Cryptographic Operations

The TOE application performs cryptographic operations for the purpose of securing local disk drive(s) using symmetric encryption techniques and for securing sessions between the TOE and the PGP Universal Server in the Operational Environment. The TOE application leverages the PGP software developer's kit cryptographic module validated to FIPS 140-2 (Cert. #1101). FIPS approved ciphers are used for disk encryption activity and SSLv3.1/TLSv1.0 sessions with the PGP Universal Server in the Operational Environment.

Cryptographic operations for the TOE application are provided by the PGP SDK subsystem, the PGP Cryptographic Engine, the Bootguard subsystem, and the BIOS Filter. The PGP SDK is a "FIPS 140-2 validated cryptographic module". The Bootguard subsystem is located within the BIOS portion of the TOE mWDE software and converts passphrases to cryptographic keys used to access Disk Encryption Keys, Link Keys and Session Keys. Passphrase crypto key access is implemented via RFC 4880. The Bootguard subsystem does not include the random number generator (RNG) portion as it operates during the pre-boot process where key generation is not required. The BIOS Filter and PGP Cryptographic Engine components both include AES engine implementations, written in "C" that support encryption/decryption operations during the pre-boot (BIOS Filter) and Post Boot (PGP Cryptographic Engine) stages.

The TOE application generates keys using a software based RNG within the PGP SDK cryptographic subsystem that complies with the ANSI X9.31 standard. When a key is created or accessed to encrypt a disk or partition, the PGP SDK cryptographic subsystem also creates a Whole Disk Recovery Token/passphrase for use in the event the User's passphrase is forgotten. The Whole Disk Recovery Token is generated regardless of whether or not there is a Universal Server deployed with the mWDE.

Following key generation, the Whole Disk Recovery Token is transferred to the PGP Universal Server in the Operational Environment for storage. This key may be accessed by the PGP Universal Server Administrator when needed to access a drive when the main passphrase is unavailable. When performing encryption of the entire physical drive or specified partitions, the TOE application exclusively utilizes the AES algorithm (alg. Cert. #1253) with a key size of 256 bits that complies with FIPS 140-2. Drive encryption operations are conducted by the PGP cryptographic engine 4.0 within the Disk Filter Driver. Decryption activities during the pre-boot stage are conducted by the BIOS Filter.

The passphrase entered during the Bootguard authentication process is held in memory only until the Bootguard converts it to a cryptographic key. Following this step, the session key is decrypted by the BIOS Filter and is read into memory. The passphrase memory location is then zeroized.

Following the startup authentication processes, the session key is cached in non-paged memory locations within the Disk Filter driver, operating in the kernel space for encryption/decryption operations, until rebooting or shutdown of the installed platform

Once created, the session key is encrypted by the PGP SDK with the hashed value of the passphrase/smartcard or token key provided during the initial TOE operation and is stored on the disk drive in this encrypted form.

The PGP SDK cryptographic subsystem also generates keys for the purposes of securing sessions between the TOE and the PGP Universal Server in the Operational Environment. These sessions are used to transfer log records from the TOE application to the PGP Universal Server and to download policy settings from the Universal Server to the TOE application. The TOE application utilizes AES (Cert. #954), 128 bit symmetric keys for encrypting these sessions through SSLv3.1 or TLSv1.0 protocols.

The following TOE components support Cryptographic Services:

- BIOS Filter (AES)

- Bootguard (AES, SHA1)
- PGP Cryptographic Engine
- PGP SDK

BIOS Filter and Bootguard don't provide user data protection, thus only algorithm certificates are needed for AES in BIOS Filter, and AES and SHA-1 in the Bootguard.

1.9.4 Security Management

The PGP TOE application provides for security management through a full featured PGP Desktop Graphical User Interface (GUI). This interface allows the user to perform encryption operations on drives or partitions, select interface options, access help screens and documentation, create and modify passphrases and configure caching options for passphrases/keys is use.

The Security Management security function is provided through the User Interface subsystem working with the PGP Engine subsystem, to access application configuration and management options. Security Management settings are stored within application files stored within the program files folder on the local Windows Operating System.

1.9.5 Protection of TOE

Protection of the TOE is provided through the encryption mechanism described in Section 1.9.3, Cryptographic Operations and by the Pre-Boot authentication processes enforced by the Bootguard subsystem. The pre-boot process effectively isolates the system disk BIOS, Operating System component and physical disk from potential malicious activities, prior to entry of a valid user generated or recovery passphrase. This pre-boot isolation is realized through the positioning of the PGP MBR in front of the system MBR in partition 0 and by making available only a 1 MB partition in unencrypted form for the purpose of hosting the Bootguard exclusively. Therefore, only the Bootguard bootloader is able to execute within the limited 1MB partition and by enforcing that a valid passphrase/key is required in order to traverse beyond the login screen during startup to the actual system MBR, the protected user data and TOE application is effectively isolated from malicious attacks.

Additionally, the protection of the TOE installation environment (including mWDE pre-boot runtime, mWDE disk session key record, mWDE user record etc) comes in two forms:

First: protection is provided by the Windows OS in the Operational Environment, the WDE environment is contained in a read-only/system/hidden file so it is not readily accessible.

Second: protection is provided by the WDE disk filter driver, which blocks reading and writing to the file that contains the TOE software environment. The WDE disk filter driver inside Windows OS cannot effectively be bypassed as any tampering of the WDE filter driver will result in OS crash and a resultant unbootable system.

1.10 TOE Description: PGP Universal Server (or Generic equivalent)

The managed aspect of the TOE relates to security management policies that are created on PGP Universal server and are provided for download to TOE installations within the network. These policies are *created* using the PGP Universal Server in the Operational Environment and are *implemented* by the TOE on the installed platform.

These features include:

- Enable/Disable Whole Disk/Partition Encryption: encryption on client machines
- Enable/Disable Whole Disk/Partition Decryption: decryption on client machines
- Automatically encrypt physical disk following mWDE (client) installation
- Force Encryption of removable USB disks (feature excluded from CC evaluated configuration at mWDE client level)
- Enable recovery tokens
- Allow/Disallow Windows Single Sign-On passwords
- Enable/Disable PGP ZIP feature on mWDE client (feature excluded from CC evaluated configuration at mWDE client level)
- Enable/Disable PGP Shred feature on mWDE client (feature excluded from CC evaluated configuration at mWDE client level)
- Activate the Authentication Failure Handling feature and specify number of failed logins that results in User defined passphrase deactivation.

The TOE also is limited to the whole disk/partition encryption feature only; therefore additional features that are part of the PGP Desktop installation are disabled through settings made on the PGP Universal Server in the Operational Environment. Guidance on the steps required to establish the Common Criteria Evaluated configuration is contained in the mWDE Common Criteria Installation guidance document.

1.10.1 mWDE Enrollment with the PGP Universal Server in the Operational Environment

The mWDE includes the option of managing multiple installations from a centralized PGP Universal or equivalent server in the Operational Environment. The following paragraph describes the enrollment process used by the TOE user to establish the account on the PGP Universal Server

> The mWDE local user enrolls to the PGP Universal Server by submitting a username/password that is authenticated by the Universal Server in the Operational Environment. Once authentication is successful a cookie is provided to the

authenticated mWDE user's session. The cookie is a proxy for the applicable credentials and allows the applicable access to Universal Server resources (i.e.: Policy Attributes for download)

The mWDE application submits the associated cookie whenever communicating with the PGP Universal Server and this matches the user with the applicable access. The cookie is stored persistently within the TOE application in such a fashion that only the mWDE local user can access it.

Losing the cookie is non-destructive, in that the mWDE user is required to provide credentials again (username/password) but no data on the server is lost in any way. (And the user is handed back a new cookie)

1.11 TOE Description: Excluded Features

The following features are excluded from the Common Criteria Evaluated configuration and therefore are not included in the evaluation:

- Excluded PGP Keys Feature (disabled in TOE) including the following:
  - PGP Keys GUI interface
  - PGP Keyring access
  - PGP Key Generation/Management
  - Creating Keypairs
  - Key Backup
  - Key Distribution
  - Import/Export of Keys
  - Deleting Keys
  - Signing/Verifying Keys
  - Revoking Keys
  - Splitting/Joining Keys (reconstruction)
  - Shamir Secret Sharing
- Excluded: Use of AD as it relates to the mWDE TOE (creation and use of any AD objects defined and used by mWDE); use in Universal Server enrollment is allowed.
- Excluded: Command Line Interface (CLI)

- Excluded: Multiple users on the installed platform (single user only allowed to use installed platform)
- Excluded: TOE application software updates including those pushed from the PGP Universal Server in the Operational Environment
- Excluded: Functionality that allows Universal Server to assign permissions to the TOE on a per-mWDE-user basis
- Excluded: mWDE (local) disk Encryption to the Universal Administrator's Smart Card Key
- Excluded: One-Time Bypass feature via Bootguard
- Excluded: PGP Zip
- Excluded: PGP Shredder
- Excluded: PGP Virtual Disk
- Excluded: USB removable flash drive encryption
- Excluded: Two-Factor Authentication Using a USB Flash Device
- Excluded: Recovery Disk (decrypts TSF protected drive)
- Excluded: PGP Netshare
- Excluded: PGP Messaging

2 Conformance Claims

The TOE is conformant with Common Criteria (CC) Version 3.1, R2 Part 2 Extended

The TOE is Common Criteria (CC) Version 3.1, R2 Part 3 Conformant at EAL 4 + augmented ALC_FLR.1

The TOE is compliant with International Interpretations with effective dates on or before: September 06, 2008.

No PP compliance claims are made for this Security Target.

3 Security Problem Definition

The TOE is intended to be used either in environments in which, at most, sensitive but unclassified information is processed. This section contains assumptions regarding the security environment and the intended usage of the TOE and threats on the TOE and the Operational Environment.

3.1 Assumptions

The following conditions are assumed to exist in the operational environment.

A.ADMIN Only a single user is allowed per installed platform, The user is assumed non-hostile, appropriately trained, and assumed to follow all administrator guidance. Universal Server Administrators in the Operational Environment are likewise non-hostile, appropriately trained and follow provided guidance.

A.AUDIT_REV A PGP Universal Server or Generic Equivalent is provided in the Operational Environment which supports the review of audit records passed to it by mWDE (TOE) installations.

A. TIME_STAMPS The Operational Environment shall provide an accurate time source for use in time stamps.

A.PHYSEC The PGP Universal Server or its equivalent in the Operational Environment is physically secure.

A.AUTH The Operational Environment shall provide an authentication server for use in TOE enrollment to Universal Server.

A.ACCESS The Operational Environment provides a TPM module, SmartCard or Token (as required) and cryptographic keypairs to support secure access to cryptographic keys used to encrypt/decrypt mWDE (TOE) protected resources. Supported types are listed in Section 1.5.2.

3.2 Threats Addressed by the TOE

The threats discussed below are addressed by the PGP® TOE. The threat agents are either unauthorized persons or external IT entities not authorized to use the TOE itself.

T.AUDACC Persons may not be accountable for the changes made to TSF settings that affect behavior of TSF functions, thus allowing an attacker to make such unauthorized TSF changes and escape detection.

T.PHY_ATTACK An Attacker may obtain physical control over the platform housing the TOE after having been in a RAM de-energized, powered down state and attempt to bypass boot processes and/or authentication mechanisms during start-up to obtain unauthorized access to the physical hard disk, specified partitions and user data stored therein.

T.PROCOM An unauthorized person or unauthorized external IT entity may be able to view, modify, and/or delete security related information that is sent between a remotely located administrator and the TOE.

T.RECOVER A user may forget the passphrase used for access to an mWDE encrypted hard drive resource.

T.SELPRO An unauthorized person may read, modify, or destroy security critical TOE configuration data.

T.UNIDENT_ACT The administrator may fail to notice potential security violations, thus limiting the administrator's ability to identify and take action against a possible security breach.

3.3 Threats to be Addressed by Operating Environment

T.MGMT_SERVER A user may forget his authentication credentials (passphrase) leading to the local machine hard drive resource (encrypted using mWDE) becoming inaccessible. In addition, Universal Server Administrators may be unable to effectively institute security policies among deployed TOE instances throughout the deployed network.

3.4 Organizational Security Policies

P.WDE_ENCRYPT The Local user and/or Universal Server Administrators must assure the platform on which the TOE is employed shall implement encryption on the Boot Partition and the underlying Operating System at a minimum.

P.PASS_POLICY The following password policy must be procedurally enforced: WDE password must be at least 8 characters, and at least one character from each of the following: capital letters, lowercase letters, numbers, and punctuation.

4 SECURITY OBJECTIVES

This section describes the security objectives for the TOE and the TOE's operating Environment. The security objectives are divided between TOE Security Objectives and Security Objectives for the Operating Environment.

The following are the IT security objectives for the TOE:

O.AUDIT_GEN The TOE shall provide the capability to generate records of selected security relevant events.

O.AUDIT_REV The TOE shall provide a means to review a readable audit trail of modifications made to attributes affecting the behavior of TSF functions and the startup of audit functions.

O.CRYPTO The TOE shall perform the following cryptographic functions: key generation, encryption, decryption, hashing, random number generation, zeroization of keys.

O.DISK_PROT The TOE shall provide mechanisms to encrypt/decrypt physical disk drives or individual partitions and thereby secure user data stored on local disk drives.

O.ENCRYPT The TOE must protect the confidentiality of data transfer with an authorized administrator (PGP Universal Server or equivalent) through encryption during sessions within the connected network.

O.IDAUTH The TOE provides mechanisms that control logical user access to the TOE and must uniquely identify and authenticate the claimed identity of the user through multiple authentication options, authentication failure handling mechanisms and obscures user password during entry before granting access to encrypted whole disk/partition on the installed platform.

O.MANAGE The TOE will provide all the functions and facilities necessary to support the local user in the management of the security of the TOE, and restrict these functions and facilities from unauthorized use.

O.RECOVER The TOE will create a recovery passphrase upon generation of a key used for Whole Disk Encryption, prevent its reuse by allowing only a "single use" (and then generating a new Whole Disk Recovery Token (WDRT)) and will send that key to the PGP Universal Server or its equivalent in the Operational Environment.

O.SELPRO The TOE must protect itself against attempts by unauthorized users to bypass, deactivate, or tamper with TOE security functions.

O.WDE_ENCRYPT The TOE will provide the capability and required guidance for the local user and/or Universal Server Administrator to ensure that, at a minimum, the Boot Partition and underlying operating system of the installed platform are encrypted using TOE mechanisms.

4.1 Security Objectives for the Environment

The following security objectives apply to the Operational Environment and are satisfied by technical means by Operational Environment hardware/software:

OE.MANAGE A PGP Universal Server or Generic Equivalent is provided in the Operational Environment and shall be capable of managing multiple TOE application installations within the deployed network including audit review, management of Whole Disk Recovery Tokens/passphrases and policy management.

OE.ACCESS The Operational Environment provides a TPM module, SmartCard or Token (as required) and cryptographic keypairs to support secure access to cryptographic keys used to encrypt/decrypt mWDE (TOE) protected resources. Supported types are listed in Section 1.5.2.

OE.AUDIT_REV The PGP Universal Server or Generic Equivalent provided in the Operational Environment supports the review of audit records passed to it by mWDE (TOE) installations.

The following are the non-IT security objectives, which, in addition to those assumptions, are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

OE.ADMIN Only a single user is allowed per installed platform. the TOE local user and PGP Universal Server Administrators are non-hostile and follow all administrator guidance; however, they are capable of error.

OE.PASS_POLICY The following password policy must be procedurally enforced: WDE password must be at least 8 characters, and at least one character from each of the following: capital letters, lowercase letters, numbers, and punctuation.

OE.PHYSEC The PGP Universal Server or its equivalent is located in a physically secure server room environment.

OE.TIME_STAMPS The Operational Environment shall provide an accurate time source for use in time stamps for audit records.

OE.AUTH The Operational Environment shall provide an authentication server for use in TOE enrollment to Universal Server.

4.2 Mapping of Security Environment to Security Objectives

The following table represents a mapping of the threats to the security objectives defined in this ST.

	O.AUDIT_GEN	O.AUDIT_REV	O.CRYPTO	O.DISK_PROT	O.ENCRYPT	O.IDAUTH	O.RECOVER	O.MANAGE	O.SELPRO	O.WDE_ENCRYPT	OE.TIME_STAMPS	OE.AUDIT_REV	OE.AUTH	OE.ACCESS	OE.MANAGE	OE.ADMIN	OE.PHYSEC	OE.PASS_POLICY
T.AUDACC	X	X									X							
T.PHY_ATTACK			X	X														
T.PROCOM			X		X	X												
T.RECOVER							X								X			
T.SELPRO						X			X									
T.UNIDENT_ACT	X	X						X			X				X			
T.MGMT_SERVER															X			
A.ADMIN																X		
A.AUDIT_REV												X						
A.AUTH													X					
A.ACCESS														X				
A.PHYSEC																	X	
A.TIME_STAMPS											X							
P.WDE_ENCRYPT										X								
P.PASS_POLICY																		X

Table 6: Summary of Mappings between Assumptions/Threats and IT/Environment Security Objectives

4.3 Rationale For IT SECURITY OBJECTIVES

T.AUDACC

O.AUDIT_GEN addresses this threat by generating audit records of audit startup and modifications in the behavior of TSF functions.

This threat is also mitigated by O.AUDIT_REV and OE.TIME_STAMPS which provides a means to generate & record a readable audit trail of security-related events, with accurate dates and times, and a means to sort the audit trail based on relevant attributes.

T.PHY_ATTACK

O.CRYPTO mitigates this threat by using cryptography for the purposes of encrypting/decrypting the physical disk drive or selected partition(s). O.DISK_PROT further mitigates this threat by providing mechanisms that protect the physical disk by encrypting the physical hard disk, or partition(s) thereby prevent unauthorized access.

T.PROCOM

O.IDAUTH addresses this threat by providing mechanisms that control logical user access to the platform encrypted physical disk drive for the installed platform (and therefore the TOE).

O.CRYPTO mitigates this threat by using cryptography for use by the TOE in securing sessions between the TOE application and the PGP Universal Server or its equivalent in the Operational Environment.

O.ENCRYPT mitigates this threat by securing data transfers between the TOE and the PGP Universal Server or its equivalent by encrypting these sessions using SSLv3.1/TLSv1.0.

T.RECOVER

O.RECOVER mitigate this threat by creating a recovery passphrase upon generation of a key used for Whole Disk/Partition Encryption and sends that key to the PGP Universal Server or its equivalent in the Operational Environment for storage.

OE.MANAGE further mitigates this threat by providing a management server in the Operational Environment for the storage of recovery passphrases.

T.SELPRO

O.SELPRO mitigates this threat by providing mechanism to protect itself against attempts by unauthorized users to bypass, deactivate, or tamper with TOE security functions. The TOE realizes this primarily through the Bootguard component enforcing authentication requirements & resisting bypass attempts.

	O.IDAUTH addresses this threat by providing mechanisms that control logical user access to the platform encrypted physical disk drive for the installed platform (and therefore the TOE).
T.UNIDENT_ACT	O.AUDIT_GEN mitigates this threat by providing the capability to generate records of security relevant events associated with users. O.AUDIT_REV further mitigates this threat by providing a means to record a readable audit trail of security-related events, with accurate dates and times, and a means to sort the audit trail based on relevant attributes. O.MANAGE further mitigates this threat by providing all the functions and facilities necessary to support the local user in the management of the security of the TOE, and restrict these functions and facilities from unauthorized use. OE.TIME_STAMPS further mitigates this threat by providing an accurate time source (time stamps) for use in audit records produced by the TOE. OE.MANAGE further mitigates this threat by providing the means for an administrator in the Operational Environment to manage multiple TOE installations using the PGP Universal Server or its equivalent.
T.MGMT_SERVER	OE.MANAGE mitigates this threat by providing a PGP Universal Server (management server) or its equivalent in the Operational Environment for the purposes of managing deployed TOE instances within the deployed environment, including TOE application audit review, passphrase recovery and policy deployment.
P.WDE_ENCRYPT	O.WDE_ENCRYPT specifies that the TOE supports the capability for the TOE user or Universal Server administrator to ensure, at a minimum, the Boot Partition and underlying Operating System of the installed platform is encrypted using TOE mechanisms, thereby, implementing the stated policy.
P.PASS_POLICY	OE.PASS_POLICY specifies that the following password policy is procedurally enforced: WDE password must be at least 8 characters, and at least one character from capital letters, lowercase letters, numbers, and punctuation.

4.4 Rationale For Assumption Coverage

This section provides a justification that for each assumption and the security objectives for the environment which cover that assumption.

A.ADMIN	The assumption A.ADMIN is addressed in the objective OE.ADMIN which ensures that only a single user is allowed per installed platform and the user and PGP Universal Server Administrators are non-hostile and follow all administrator guidance.
A.PHYSEC	The assumption A.PHYSEC is addressed in the objective OE.PHYSEC which specifies that the PGP Universal Server or its equivalent is located in a physically secure server room environment.
A.TIME_STAMPS	The assumption A.TIME_STAMPS is restated directly in the objective OE.TIME_STAMPS which specifies that the Operational Environment shall provide an accurate time source for use in time stamps for audit records.
A.AUTH	The assumption A.AUTH is restated directly in the objective OE.AUTH which specifies that the Operational Environment shall provide an authentication server for use by the TOE in Universal Server enrollment.
A.ACCESS	The assumption A.ACCESS is restated directly in the objective OE.ACCESS which specifies that the Operational Environment provides a TPM module, SmartCard or Token (as required) and cryptographic keypairs to support secure access to cryptographic keys used to encrypt/decrypt mWDE (TOE) protected resources.
A.AUDIT_REV	The assumption A.AUDIT_REV is restated directly in the objective OE.AUDIT_REV which specifies that the Operational Environment provides the ability to review TOE audit records using the Universal Server or Equivalent.

5 Extended Components Definition

Explicit TOE Security Functional Requirements	
FAU_GEN_EXP.1	Audit Generation - Explicit
FCS_COP_EXP.1	Cryptographic operation- Recovery Passphrase Generation

Table 7: Extended Components - Explicit SFRs

5.1 Explicitly Stated TOE Security Functional Requirements

The SFRs defined in this section are explicitly stated and are derived from similar requirements in Part 2 of the CC.

5.1.1 Class FAU: Security Audit-Explicit

5.1.1.1 FAU_GEN_EXP.1 – Audit Generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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

- ***Start-up and*** ~~***shutdown of the audit functions***~~;
- All auditable events for the not specified level of audit; and
- **Events listed in Table 8: Audited Events**

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

• Date and time of the event, type of event, ~~***subject identity***~~, and the outcome (success or failure) of the event; and

• For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **no other audit relevant information**.

Component	Event	Details
FAU_GEN_EXP.1*	Audit logging startup at specified level (enable)	Logging set to: “Normal” (default)

FIA_UID.2	All use of the user identification mechanism	Success/Failure of Identification
FIA_UAU.2	All use of the user authentication mechanism	Success/Failure of Authentication
FMT_MOF.1a*	All modifications in the behavior of the functions of the TSF	Security function modification affecting behavior

Table 8: Audited Events

*note: audit enable is logged and the logging level specified, however, audit disable is not logged locally, but is logged via XML to the PGP Universal Server or its equivalent in the Operational Environment – some actions activated via the Universal Server may be logged to the Universal Server resource.

5.1.2 Class FCS: Cryptographic Support - Explicit

5.1.2.1 FCS_COP_EXP.1 Cryptographic operation- Recovery Passphrase Generation

Hierarchical to: No other components.

Dependencies: None

FCS_COP_EXP.1.1 The TSF shall perform Recovery Passphrase generation without user intervention for each of the following operations:

- Encryption of physical drive (whole disk or partition)
- Encryption of physical drive partition
- Use of recovery passphrase (“single use” only allowed)

FCS_COP_EXP.1.2 Recovery Passphrase generation shall be in accordance with a specified cryptographic algorithm AES, with Base32 encoding and cryptographic key sizes 128,256 bit that meet the following: FIPS SP 800-90, ANSI X9.31 (AES), RFC 3548 (Base32).
(AES Cert. #954)

FCS_COP_EXP.1.3 Following the creation of the Recovery Passphrase, the (mWDE) TSF shall transfer the Recovery Passphrase to the PGP Universal

Server or its equivalent in the Operational Environment.

5.2 Rationale for Explicitly Stated Security Requirements

Table 9: Explicitly Stated SFR Rationale below presents the rationale for the inclusion of the explicit requirements found in this Security Target.

Explicit Requirement	Identifier	Rationale
FAU_GEN_EXP.1	Audit Generation-Explicit	This SFR needs to be explicitly stated as the SFR (FAU_GEN.1) requires that the TOE generate audit logs for startup and shutdown of the audit function and includes the subject identity in audit records. The TOE does not produce audit records for shutdown activities and does not include subject identity in audit records.
FCS_COP_EXP.1	Cryptographic operation-Recovery Passphrase Generation	This SFR needs to be explicit in that there are no SFRs available to define the automatic generation of Whole Disk Recovery Tokens for Whole Disk/Partition encryption operations executed by the TOE.

Table 9: Explicitly Stated SFR Rationale

6 Security Requirements

The security requirements that are levied on the TOE are specified in this section of the ST. These security requirements are defined in Sections 6.1

TOE Security Functional Requirements (from CC Part 2)	
FAU_SAR.1	Audit Review
FAU_SAR.3	Selectable Audit Review
FCS_CKM.1a	Cryptographic Key Generation – *software RNG (disk encryption)*
FCS_CKM.1b	Cryptographic Key Generation – *software RNG (SSLv3.1/TLSv1.0)*
FCS_CKM.1c	Cryptographic Key Generation – *S2K*
FCS_CKM.3	Cryptographic Key Access
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1a	Cryptographic Operation*: Whole Disk, Partition*
FCS_COP.1b	Cryptographic Operation*: SSLv3.1/TLSv1.0 Client to Universal Server session encryption*
FCS_COP.1c	Cryptographic Operation*: Hashing*
FIA_AFL.1	Authentication Failure Handling
FIA_UID.2	User Identification Before Any Action
FIA_UAU.2	User Authentication Before Any Action
FIA_UAU.5	Multiple Authentication Mechanisms
FIA_UAU.7	Protected Authentication Feedback
FMT_MOF.1a	Management Of Security Function behavior
FMT_MOF.1b	Management Of Security Function behavior *from Universal Server*
FMT_MTD.1	Management Of TSF Data - *Create, Modify, Delete*
FMT_SMF.1	Specification Of Management Functions
FPT_ITI.1	Inter-TSF Detection Of Modification

Table 10: Functional Requirements

6.1 TOE Security Functional Requirements

The SFRs defined in this section are taken from Part 2 of the CC.

6.1.1 Class FAU: Security Audit

6.1.1.1 FAU_SAR.1 Audit Review

FAU_SAR.1.1 The TSF shall provide **any "post-boot" user** with the capability to read **audit startup and modifications in the behavior of the functions of the TSF** 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.

6.1.1.2 FAU_SAR.3 Selectable Audit Review

FAU_SAR.3.1 The TSF shall provide the ability to ***perform*** **sorting** of audit data based on **Level (Verbose, Info, Error, Warn), Date (today, yesterday, 2-7 days ago), Topic (All, PGP, IM, Email, Whole Disk).**

6.1.2 Class FCS: Cryptographic Support

6.1.2.1 FCS_CKM.1a Cryptographic Key Generation – *software RNG (disk encryption)*

FCS_CKM.1.1a The TSF shall generate ***symmetric*** cryptographic keys ***for whole disk encryption*** in accordance with a specified cryptographic key generation algorithm **FIPS 140-2 (Cert. #1101) Validated Software RNG (RNG Cert. #539) in the PGP SDK cryptographic module subsystem using 3DES (Certs. #754 and #895)** and specified cryptographic key sizes **256 bits (AES)** that meet the following: **FIPS 140-2.**

6.1.2.2 FCS_CKM.1b Cryptographic Key Generation – *software RNG (SSLv3.1/TLSv1.0)*

FCS_CKM.1.1b The TSF shall generate ***symmetric*** cryptographic keys ***for SSLv3.1/TLSv1.0 sessions*** in accordance with a specified cryptographic key generation algorithm **FIPS 140-2 (Cert. #1101) Validated Software RNG (RNG Cert. #539) in the PGP SDK cryptographic subsystem using 3DES (Certs. #754 and #895)** and specified cryptographic key sizes **128, 256 bits (AES), 192 bits (3DES)** that meet the following: **FIPS 140-2.**

6.1.2.3 FCS_CKM.1c Cryptographic Key Generation – S2K

FCS_CKM.1.1c The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm **PGP string to key (S2K) operation using SHA1 (Cert. #1203) with 16 bit salt** and specified cryptographic key sizes **none** that meet the following: **OpenPGP RFC 4880.**

6.1.2.4 FCS_CKM.3 Cryptographic key access

FCS_CKM.3.1 The TSF shall perform **Disk Access Key Encryption/Decryption** in accordance with a specified cryptographic key access method

- **User Defined Passphrase**

that meets the following:

Passphrase per OpenPGP RFC 4880* (iterated salted string to key using 16 bytes), pad per PKCS#1, encrypt per (AES 256 (Certs. #954, #1253, #1316 and #1317)) FIPS 140-2;

*SHA1 used for S2K (RFC 4880) validated per Cert. #1203)

This operation performed by the Bootguard during pre-boot and by the SDK subsystem post boot.

6.1.2.5 FCS_CKM.4 Cryptographic key destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method **zeroization** that meets the following: **FIPS 140-2.**

6.1.2.6 FCS_COP.1a Cryptographic operation: *Whole Disk, Partition*

FCS_COP.1.1a The TSF shall perform **Whole Disk or Partition Encryption/Decryption** in accordance with a specified cryptographic algorithm **AES (Certs. #954, #1253, #1316 and #1317)** and cryptographic key sizes **256 bit** that meet the following: **RFC 3565 (AES), FIPS PUB 140-2**.

Encryption/Decryption is performed by the AES implementation within the Disk Filter Driver/PGP Cryptographic Engine. Disk Decryption during pre-boot performed by an AES implementation within the BIOS Filter.

6.1.2.7 FCS_COP.1b Cryptographic operation: SSLv3.1/TLSv1.0 Client to Universal Server session encryption

FCS_COP.1.1b The TSF shall perform **SSLv3.1/TLSv1.0 session encryption/decryption** in accordance with a specified cryptographic algorithm **AES (Certs. #954, #1253, #1316 and #1317), 3DES (Certs. #754 and #895) using RSA/DH (RSA Cert. #460)** and cryptographic key sizes **128 bit (AES), 192 bits (3DES) 512, 1024, 2048 (RSA/DH) that** meet the following: **FIPS 140-2**

6.1.2.8 FCS_COP.1c Cryptographic operation: *Hashing*

FCS_COP.1.1c The TSF shall perform **Hashing** in accordance with a specified cryptographic algorithm **SHA-1, SHA-2 (Certs. #926, #1149 and #1203)** and cryptographic key sizes **160 (SHA-1)**, **256, 384, 512 (SHA-2) bits** that meet the following: **FIPS 140-2, FIPS 180-2**

6.1.3 Class FIA: Identification and Authentication

6.1.3.1 FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall detect when a ***PGP Universal Server*** configurable positive integer within **3 – 99 of** unsuccessful authentication attempts occur related to **User passphrase based* authentication/decryption during system boot (bootguard).**

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met the TSF shall **disable the User passphrase***

*applies only when passphrase authentication is used, this disable action does not affect the WDRT, token or TPM accounts therefore once a valid WDRT, token or TPM passphrase is entered, all passphrase users are also unlocked.

6.1.3.2 FIA_UAU.2 User authentication before any action

FIA_UAU.2.1 The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.

6.1.3.3 FIA_UAU.5 Multiple authentication mechanisms

FIA_UAU.5.1 The TSF shall provide

- **User defined passphrase (BIOS log in dialog requests passphrase):**
  - **When invoked: Upon platform boot (bootguard)**
  - **Requirements for success: The User provided passphrase**

when hashed must match the passphrase (used during disk encryption) used to encrypt the Disk Access Key and therefore decrypt the Disk Access Key.

- **Smartcard or Token-Based Authentication (BIOS log in dialog requests token for authentication):**
  - o **When invoked: Upon platform boot (bootguard)**
  - o **Requirements for success: The correct public Key used to encrypt the Disk Access key must be presented to the platform from the applicable Token Device and therefore decrypt the Disk Access Key.**
- **Trusted Platform Module (BIOS log in dialog requests passphrase)**
  - o **When invoked: Upon platform boot (bootguard)**
  - o **Requirements for success: The correct User provided passphrase when hashed must match the hashed value of the passphrase (used during disk encryption) along with the Storage Root key of the TPM used to encrypt the Disk Access Key and therefore decrypt the Disk Access Key.**
- **Universal Server authentication to mWDE (TOE) application**
  - o **When invoked: Upon communicating with the Universal Server for the purpose of downloading policies, uploading audit records**
  - o **Requirements for success: The Universal Server must present to the TOE application a trusted server certificate as a prerequisite to establishing a SSLv3.1/TLSv1.0 session.**

to support user authentication.

FIA_UAU.5.2 The TSF shall authenticate any user's claimed identity according to the **Applicable User's selection of method used to encrypt the hard disk/partition; Universal Server: Presentation of a valid trusted certificate.**

6.1.3.4 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.

6.1.3.5 FIA_UID.2 User identification before any action

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

6.1.4 Class FMT: Security Management

6.1.4.1 FMT_MOF.1a Management of Security Function behavior

FMT_MOF.1.1a The TSF shall restrict the ability to enable, disable **Whole Disk/Partition Encryption*** to **the locally authenticated User**.

* enabling/disabling Whole Disk/Partition Encryption in this instance is encryption/decryption of the Whole Disk/Partition.

6.1.4.2 FMT_MOF.1b Management of Security Function behavior – *from Universal Server*

FMT_MOF.1.1b The TSF shall restrict the ability to enable, disable:

- **Cryptographic Operations**
  - **User-Initiated Whole Disk/Partition Encryption (allow/disallow user-initiated encryption)**
    via policy from Universal Server
  - **User-Initiated Whole Disk/Partition Decryption (allow/disallow removal of encryption)**
    via policy from Universal Server
  - **Recovery Tokens**
    via policy from Universal Server
- **Authentication**
  - **Recovery tokens**
    via policy from Universal Server
  - **Allow/Disallow Windows Single Sign-On**
    via policy from Universal Server
  - **Activate the Authentication Failure Handling feature and specify number of failed logins that results in User defined passphrase deactivation**
    via policy from Universal Server

to **the authorized PGP Universal Server***

*The Universal Server component or its equivalent is part of the Operational Environment.

6.1.4.3 FMT_MTD.1 Management of TSF Data-*Create, Modify, Delete*

FMT_MTD.1.1 The TSF shall restrict the ability to create, modify, delete the

- **Passphrase associated with the Physical disk or Partition encryption keys**
- **TOE Accounts (adding, modifying, removing users)**

to **the locally authenticated user.**

6.1.4.4 FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- **Encrypt/Re-encrypt Physical Disk, Disk Partition(s)**
- **Decrypt Physical Disk, Disk Partition(s)**
- **Modify encryption keys**
- **Review/Filter Audit logs**
- **Create/Modify Passphrase**
- **Create/Modify/Delete users (assumes single user is retained)**
- **Enable/Disable audit logging**
- **Implement Universal Server Policy: Enable/Disable Whole Disk/Partition Encryption**
- **Implement Universal Server Policy: Automatically encrypt physical disk following installation**
- **Implement Universal Server Policy: Enable/Disable Recovery tokens**
- **Implement Universal Server Policy: Allow/Disallow Windows Single Sign-On**
- **Implement Universal Server Policy: Activate the Authentication Failure Handling feature and specify number of failed logins that**

results in User defined passphrase deactivation

6.1.5 Class FPT: Protection of the TSF

6.1.5.1 FPT_ITI.1 Inter-TSF detection of modification

FPT_ITI.1.1 The TSF shall provide the capability to detect modification of all TSF data during transmission between the ***(mWDE)*** TSF and another trusted IT product within the following metric: **a single Message Authentication Code (MAC) error during transmission.**

FPT_ITI.1.2 The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the ***(mWDE)*** TSF and another trusted IT product and ***~~perform~~ request*** **resending of transmitted data** if modifications are detected.*

* the mWDE requests the Universal Server or equivalent to resend data if modifications are detected.

6.2 Rationale For TOE Security Requirements

	O.AUDIT_GEN	O.AUDIT_REV	O.CRYPTO	O.DISK_PROT	O.ENCRYPT	O.IDAUTH	O.RECOVER	O.MANAGE	O.SELPRO	O.WDE_ENCRYPT
FAU_GEN_EXP.1	X									
FCS_COP_EXP.1							X			
FAU_SAR.1		X								
FAU_SAR.3		X								
FCS_CKM.1a,b,c			X							
FCS_CKM.3			X							
FCS_CKM.4			X							
FCS_COP.1a			X	X					X	X
FCS_COP.1b			X		X					
FCS_COP.1c			X							
FIA_AFL.1						X				
FIA_UID.2						X				
FIA_UAU.2						X				
FIA_UAU.5						X				
FIA_UAU.7						X				
FMT_MOF.1a,b								X		

FMT_MTD.1								X		
FMT_SMF.1								X		
FPT_ITI.1					X				X	

Table 11: Summary of Mappings between Security Functional Requirements and IT Security Objectives

6.2.1 TOE Security Functional Requirements Rationale

The security requirements are derived according to the general model presented in Part 1 of the Common Criteria. Specifically, Table 11: Summary of Mappings between Security Functional Requirements and IT Security Objectives illustrates the mapping between the security requirements and the security objectives and Table 6: Summary of Mappings between Assumptions/Threats and IT/Environment Security Objectives demonstrates the relationship between the threats, policies and IT security objectives. The functional and assurance requirements presented in this Security Target are mutually supportive and their combination meets the stated security objectives.

Security Objective	Mapping Rationale
O.AUDIT_GEN	FAU_GEN_EXP.1 specifies that the TSF generates audit records for security related events and includes date/time, type of event, outcome and details listed in Table 8: Audited Events
O.AUDIT_REV	FAU_SAR.1 specifies that the TSF provides the capability for the any post boot user to read audit records and presents the data in a suitable manner. FAU_SAR.3 specifies that the TSF provides the ability to sort audit data based on Message and Time categories.
O.CRYPTO	FCS_CKM.1a specifies that the TSF utilizes a software RNG to generate symmetric keys for Whole disk encryption using the specified algorithms and keys sizes. FCS_CKM.1b specifies that the TSF utilizes a software RNG to generate symmetric keys for SSLv3.1/TLSv1.0 sessions using the specified algorithms and keys sizes. FCS_CKM.1c specifies that the TSF executes a String to Key operation by hashing the entered passphrase using SHA1 and applying a 16 bit salt in accordance with RFC 4880. FCS_CKM.3 specifies that the disk access key used for disk encryption may be accessed via User Passphrase, Smartcard or Token or TPM based on the selection made at the time of disk encryption. FCS_CKM.4 specifies that the TSF destroys keys through zeroization that meets FIPS 140-2. FCS_COP.1a specifies that the TSF performs Whole Disk or Partition(s) Encryption using

	the AES algorithm and 256 bit keys that meet FIPS 140-2. FCS_COP.1b specifies that the TSF performs encryption of TOE (Client) to Universal Server sessions using the SSLv3.1/TLSv1.0 protocol and the specified algorithms and key sizes. FCS_COP.1c specifies that the TSF performs hashing using the SHA-1, SHA-2 algorithm and specified key sizes that meet FIPS 180-1 (SHA-2).
O.DISK_PROT	FCS_COP.1a specifies that the TSF protects data on the physical disk drives by performing Whole Disk or Partition(s) Encryption using the AES algorithm and 256 bit keys that meet FIPS 140-2.
O.ENCRYPT	FCS_COP.1b specifies that the TSF protects the confidentiality of data transfer between the TOE application and the PGP Universal Server in the Operational Environment by the encryption of mWDE Client to Universal Server sessions using SSLv3.1/TLSv1.0 and the specified algorithms and key sizes.
O.IDAUTH	FIA_AFL.1 specifies that the TSF will disable the User Passphrase account upon exceeding the PGP Universal Server Administrator's configured number of failed logins. FIA_UID.2 specifies that the TSF controls logical access to the TOE by requiring that each user is identified before allowing TSF mediated actions on behalf of that user. FIA_UAU.2 specifies that the TSF controls logical access to the TOE by requiring that each user is authenticated before allowing TSF mediated actions on behalf of that user. FIA_UAU.5 specifies that the TSF provides for multiple authentication methods including: Passphrase, Smartcard or Token (public key) or TPM (Passphrase). FIA_UAU.7 specifies that the TSF obscures feedback to the user when passwords (authentication data) are entered.
O.RECOVER	FCS_COP_EXP.1 specifies that the TSF generates a Recovery Passphrase for Physical Drive encryption or Physical Partition(s) encryption.
O.MANAGE	FMT_MTD.1 specifies that the TSF restricts the ability to create, modify, and delete the listed TSF data types. FMT_SMF.1 specifies that the TSF provides the listed management functions, including those applied from the Universal Server. FMT_MOF.1a specifies the conditions under which the local user can modify TSF behavior relating to encryption/decryption. FMT_MOF.1b specifies that the Universal Server Administrator has exclusively access to modify polices that impact local TOE implementation of security functions.
O.SELPRO	FCS_COP.1a specifies that the TSF protects itself against attempts by unauthorized users to bypass, deactivate or tamper with TOE security functions by performing Whole Disk or Partition(s) Encryption using the AES algorithm and 256 bit keys that meet FIPS 140-2 (and protects the TSF via disk encryption) FPT_ITI.1 specifies that the TSF protects itself against attempts by unauthorized users to bypass, deactivate or tamper with TOE security functions when accessed remotely by providing the capability to detect modification of TSF data during transmission between the TOE application and a trusted IT product in the Operational Environment. The TSF also provides the capability to verify integrity for these transmissions and requests

	resending of transmitted data in the event modifications are detected.
O.WDE_ENCRYPT	FCS_COP.1a specifies that the TSF protects data on the physical disk drives by performing Whole Disk or Partition(s) Encryption using the AES algorithm and 256 bit keys that meet FIPS 140-2.

6.3 Rationale For IT Security Requirement Dependencies

This section includes a table of all the security functional requirements and their dependencies and a rationale for any dependencies that are not satisfied.

Functional Component	Dependency	Included/Rationale
FAU_GEN_EXP.1	FPT_STM.1	No
FCS_COP_EXP.1	FCS_COP.1a, FCS_CKM.4	Yes
FAU_SAR.1	FAU_GEN.1	Yes
FAU_SAR.3	FAU_SAR.1	Yes
FCS_CKM.1a	FCS_COP.1a, FCS_CKM.4	Yes
FCS_CKM.1b	FCS_COP.1b, FCS_CKM.4	Yes
FCS_CKM.1c	FCS_COP.1b, FCS_CKM.4	Yes
FCS_CKM.3	FCS_CKM.1c, FCS_CKM.4	No, FCS_CKM.4
FCS_CKM.4	FCS_CKM.1a,b,c	Yes
FCS_COP.1a	FCS_CKM.1a, FCS_CKM.4	Yes
FCS_COP.1b	FCS_CKM.1b, FCS_CKM.4	Yes
FCS_COP.1c	FCS_CKM.1b, FCS_CKM.4	Yes
FIA_AFL.1	FIA_UID.1	Yes* via FIA_UID.2
FIA_UID.2	None	N/A
FIA_UAU.2	FIA_UID.1	Yes* via FIA_UID.2
FIA_UAU.5	None	N/A
FIA_UAU.7	FIA_UAU.1	Yes* via FIA_UAU.2
FMT_MOF.1a,b	FMT_SMR.1, FMT_SMF.1	No* FMT_SMR.1
FMT_MTD.1	FMT_SMR.1, FMT_SMF.1	No* FMT_SMR.1
FMT_SMF.1	None	None

Functional Component	Dependency	Included/Rationale
FPT_ITI.1	None	None

Table 12: SFR Dependencies

6.4 Rationale For IT Security Requirement Dependencies not satisfied

The following rationale is provided for those dependencies that are not satisfied within this Security Target:

6.4.1 FPT_STM.1 Time Stamps dependency to FAU_GEN_EXP.1

The requirement for FPT_STM.1 Time Stamps is not satisfied by the TOE as the TOE does not have mechanisms for maintaining or managing a time source within the application. The TSF does support the use of Time Stamps in audit records by leveraging the time source mechanism provided through the underlying Operating System. Therefore, this dependency is satisfied through security objective OE.Time_Stamps in the Operational Environment.

6.4.2 FMT_SMR.1 Security Roles dependency to FMT_MOF.1a, b; FMT_MTD.1

The requirement for FMT_SMR.1 is not satisfied by the TOE as the TOE does not include mechanisms for assigning roles. There is no distinction by role for cryptographic access. The TOE manages user access exclusively through key assignments made during encryption activities, if the user holds a key used to encrypt a particular resource and enters the required passphrase to release that key; the user may access the resource.

Options may be assigned to specific users allowing or restricting use of particular security functions within the TOE, however, this is done through settings made within the PGP Universal Server in the Operational Environment. For example, one user on a given platform may be able to decrypt while another user on a separate platform cannot (function disabled) – these are assigned on a user/platform basis.

6.4.3 FCS_CKM.3 Cryptographic Key Access dependency to FCS_CKM.4 Cryptographic Key Destruction

The FCS_CKM.3 requirement is satisfied by the Bootguard subsystem that is run during the Bootstrap process and by the PGP Cryptographic Engine post boot. It does not include the capability to zeroize cryptographic keys while running in the disk bios. FCS_CKM.3 specifies 3 methods of release the Disk Access Key: Passphrase, Smartcard or Token and TPM based methods. In all these cases once the Disk Access Key is released by the TOE, the boot process continues and the Key is passed along to the application as the drive boots. The PGP Cryptographic Engine handles all Cryptographic operations once the workstation has booted and satisfies FCS_CKM.4 with a zeroization capability that is exercised during various stages of operation.

6.5 TOE Security Assurance Requirements

The assurance security requirements for this Security Target are taken from Part 3 of the CC. These assurance requirements compose an Evaluation Assurance Level 4 augmented (EAL 4 + ALC_FLR.1) as defined by the CC. The assurance components are summarized in the following table.

Assurance Class	Assurance components
ADV: Development	ADV_ARC.1 Security architecture description
	ADV_FSP.4 Complete functional specification
	ADV_IMP.1 Implementation representation of the TSF
	ADV_TDS.3 Basic modular design
AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
ALC: Life-cycle support	ALC_CMC.4 Production support, acceptance procedures and automation
	ALC_CMS.4 Problem tracking CM coverage
	ALC_DEL.1 Delivery procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_TAT.1 Well-defined development tools
	ALC_FLR.1 Basic Flaw Remediation
ASE: Security Target evaluation	ASE_CCL.1 Conformance claims
	ASE_ECD.1 Extended components definition

	ASE_INT.1 ST introduction
	ASE_OBJ.2 Security objectives
	ASE_REQ.2 Derived security requirements
	ASE_SPD.1 Security problem definition
	ASE_TSS.1 TOE summary specification
ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.2 Testing: security enforcing modules
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.3 Focused vulnerability analysis

Table 13: Assurance Requirements: EAL 4 + ALC_FLR.1

6.5.1 TOE Security Assurance Requirements

EAL 4 + ALC_FLR.1 was chosen to provide a “enhanced basic” level of independently assured security. The chosen assurance level is consistent with the threat environment. Specifically, that the threat of malicious attacks is not greater than “enhanced basic” and the product will have undergone a search for obvious flaws and a focused vulnerability analysis.

7 TOE Summary Specification

7.1 TOE Security Functions

The TOE's security functionality is organized by the following 5 Security Functions:

- Security Audit
- Identification and Authentication
- Cryptographic Operations
- Security Management
- Protection of the TOE

7.1.1 Security Audit

PGP® mWDE Audit Generation – FAU_GEN_EXP.1

The PGP TOE application creates audit records for selected security related events through the Bootguard, and the User Interface (UI) subsystems within the PGP TOE application. The TOE detects audit events exclusively based on user initiated actions. These can occur during the Pre-Boot stage, where the Bootguard subsystem generates events during the user authentication process used to access the TSF protected disk drive or during the Post-Boot stage where audit events are generated by user action taken through the UI subsystem (GUI). Any post-boot user may access audit records using the TOE application.

During the Pre-Boot stage, audit events are generated by the Bootguard subsystem, which passes the events to the Disk Filter driver subsystem to be queued pending platform startup. Once the platform has started, the Disk Filter driver signals the PGPtray (component of the WDE Engine) to access the events and stores them in the applicable application files for integration into the log file for the application and a ClientLib component (part of the WDE Engine subsystem) passes the event to the PGP Universal Server in the Operational Environment.

Following the boot up of the platform during the Post-Boot stage, audit events are detected and generated by the UI subsystem based on user initiated actions. The UI subsystem generates the event and passes it to the Disk Filter Driver. The driver then signals the PGPtray application and the event is stored in local application log files and is forwarded to the PGP Universal Server in the Operational Environment in the same method noted above for the pre-boot state.

When the audit logs are generated on the TOE, they are stored in a binary format. This format is then digested after booting and sent to the PGP Universal Server in a SOAP message (which is an XML format). In the event the Universal Server is unavailable, these logs are stored locally indefinitely pending connection with the Universal Server for transport. Locally, audit logs are stored in the \Documents and Settings\user\Application Data\PGP Corporation\PGP folder.

Auditing can be enabled/disabled locally through the GUI management interface by any post-boot user. This only affects the local logging of audit events, as the XML based audit events are generated and passed to the PGP Universal Server regardless of the “enable logging” checkbox setting within the local TOE application.

Audit logs stored locally within the TOE application files are saved for seven (7) days and then are overwritten oldest record first. The log for the current day is named PGPlog.txt. The logs for previous days are PGPlogN.txt where 'N' is a number from 1 to 7. The log files are rotated daily so that PGPlog1.txt is the previous day’s file etc.

Export of logs to the PGP Universal Server

The PGP TOE may export logs in XML format to the PGP Universal Server in the Operational Environment for compilation and review. This configuration option is contained in policies applied to the TOE application from the PGP Universal Server. The TOE application (ClientLib module within the WDE Engine) uploads the log file to a specified location, restricted based on User Identification (determined via a cookie stored on the TOE), on the PGP Universal Server.

Review of Audit Logs – FAU_SAR.1, FAU_SAR.3

There are 4 levels of logging available for viewing audit logs: Verbose, Info, Warning, Error. The application by default, logs events at the most detailed level and the logging levels relate to the viewing filters that may be applied. In addition, logs may be filtered by the component of the PGP desktop application that relates to the logged event under the “Topic” category including: All, PGP, Email, IM, Whole Disk. The Whole Disk and PGP categories relate to the core Whole Disk Encryption functionality of the TOE application. The remaining categories apply to features not enabled for the Common Criteria Evaluated configuration. The TOE only allows the review of audit records associated with modifications made affecting behavior of TSF functions and the startup of the audit function on the local platform. All other audit records must be reviewed using the Universal Server in the Operational Environment.

The audit log review functions are supported by the User Interface subsystem which provides the management GUI, including the “PGP Desktop Log” which appears as a pop up window when selected from the pull down menu. This pop window that provides the audit review interface is launched by win32 calls to the UI subsystem. The GUI interface is constructed as a Win32 application and audit logs are displayed through a Microsoft RTF widget incorporated into the UI subsystem.

The log results are stored in the rich text format (RTF) and when rendered on to the pop up log viewing screen can be filtered by either logging level or by topic as described above. When filters are applied, the UI parses the log file and only displays those lines in the file that match the filter.

7.1.2 Identification and Authentication

The TOE, with whole disk or partition encryption, requires identification and authentication prior to allowing the installed platform to be booted.

The Identification and Authentication security function operates during the pre-boot process within the TOE. During the initial boot up process, the Bootguard subsystem presents a logon interface to facilitate the entry of the applicable passphrase or smartcard or token key initial decryption of the physical drive, protected through the encryption provided by whole disk encryption. The Bootguard subsystem works in conjunction with the PGP Bootguard to hash the entered passphrase/smartcard or token key and decrypt/release the associated key material into the Bootguard application cache. When the boot process is complete and the WDE Engine is loaded, the key is passed to the WDE Engine and is stored by the Disk Filter driver in volatile memory (at the kernel level). Identification & Authentication at this level is supported by the Bootguard subsystem, which facilitates the passphrase/smartcard or token key entry interface in association with the PGP Bootguard subsystem.

Supported Authentication Methods during platform startup

The TOE supports multiple authentication methods including: Passphrase, Windows Single Sign-on, Smartcard or Token based and Trusted Platform module authentication. The TOE only allows the configuration of the application to use one of these secure authentication methods to access the disk once encrypted. Passphrase based authentication encrypts the Disk Access Key which, in conjunction with the Link key, is used to encrypt the session key used during whole disk/partition encryption. Upon entry of the passphrase the passphrase is hashed and used to decrypt the Disk Access Key used for encryption. The session key is subsequently released and read into the volatile memory for decryption of the physical disk or partition. Windows single sign-on simply utilizes the password used for Windows login for the platform and applies it in the same manner as the Passphrase authentication example. Additional details regarding the cryptography used in this process is described below in Section 7.1.3: Cryptographic Key Access – FCS_CKM.3

Smartcard or Token based authentication allows the use of a separate hardware/software token device to authenticate to the TOE and release the keys required for encryption/decryption operations. The Bootguard subsystem contains the smartcard drivers for supporting smartcard or token devices and allows the Bootguard to access key material stored on smartcards and/or tokens approved for use with the TOE. Additional details regarding the cryptography used in this process is described below in Section 7.1.3: Cryptographic Key Access – FCS_CKM.3

The Trusted platform authentication method locks the platform to the disk itself and relies on this module to orchestrate the authentication process. The TPM locking occurs through sealing the payload with the TPM's Storage Root Key along with the applicable password processed with a SHA-1 Hash. The authentication process operates in the same manner as described above except for requiring that credentials are managed via a Trusted Platform Module (TPM).

Additional details regarding the cryptography used in this process is described below in Section 7.1.3: Cryptographic Key Access – FCS_CKM.3

Identification & Authentication – FIA_UID.2, FIA_UAU.2, FIA_UAU.5, FIA_UAU.7

Passphrase and Single Sign-On Authentication

When executing an encryption operation, such as whole disk or partition encryption, the UI presents a GUI page that requests that an authentication mechanism be selected. Once the passphrase authentication option is selected, the WDE Engine subsystem requests, through the UI security management interface, that a passphrase be created by the authorized user for the purpose of authenticating to the TOE. The passphrase, once created, is hashed by the PGP SDK cryptographic subsystem and is used to wrap the Disk Access Key which ultimately is used to encrypt the Session key, which is then stored on the disk. When this passphrase is entered during the startup process, it results in the decryption of the Disk Access Keys, Link Key and finally the Session Key which allows access to the physical disk/partition.

When the Windows single sign-on option is selected, the PGP TOE application requests the password used for Windows account authentication from the authorized user verifies it against the stored Windows password and uses that password as the passphrase for Whole Disk Encryption. The Windows passphrase is then hashed by the PGP SDK subsystem and is used to encrypt the encryption key used for disk encryption in the same manner as the Passphrase option.

During the pre-boot process the applicable passphrase/smartcard or token key is requested by the Bootguard subsystem, passed to the Bootguard subsystem, hashed and used to decrypt the Disk Access Key. Following the release of the Disk Access Key and subsequent release of the Link and Session Keys, the (Windows) passphrase is still cached in volatile memory by the Bootguard subsystem and is passed to the Disk Filter subsystem when started. The Disk Filter holds the Windows passphrase until the Windows login process queries the kernel to determine if auto login is enabled. The Disk Filter driver, operating at kernel level, indicates to the Windows Operating System when queried that the auto login is enabled and then supplies the Windows passphrase to the Operating System to complete the Windows account login. The Windows passphrase is then destroyed through zeroization as the memory location is overwritten with random data (the symmetric key used for disk encryption still remains in the memory by the Disk Filter subsystem for decryption operations). This results in a single passphrase entry during the Bootguard process to result in both startup through the boot process and Windows account logon for the platform.

Identification & Authentication during boot:

During the initial platform boot process the Bootguard subsystem presents the login dialog and the authorized user enters his passphrase and/or smartcard or token key. When the passphrase is entered the characters are obscured by the TOE application to prevent observation by a casual observer. The TOE includes drivers for supported hardware/software tokens to support accessing keys as required for smartcard or token based authentication. The passphrase/smartcard or token key is hashed by the Bootguard subsystem to form a key used ultimately to decrypt the session key stored within the Bootguard keystore. I/O operations during this pre-boot stage are managed by the BIOS filter subsystem.

Authentication post boot – application management:

The TOE requires authentication credentials be entered for a limited set of post boot operations including: disk decryption, re-encryption of a disk resource, adding or removing users, and changing the user passphrase.

Authentication Failure Handling - FIA_AFL.1

The TOE enforces a limited number of failed authentication attempts during the Bootguard directed pre-boot authentication process when passphrase authentication is used. The policy setting that defines this authentication failure threshold is passed from the PGP Universal Server to the TOE (client) where it is implemented by the Bootguard subsystem. Once the user defined passphrase is disabled by the Bootguard subsystem due to excessive failed logins, the physical disk may only be decrypted using the WDRT, TPM or token passphrase. This disable feature does not apply to smartcard or token or TPM based authentication methods.

Recovery Passphrase – FCS_COP_EXP.1

The TOE utilizes a Recovery passphrase for the purpose of providing the local user an alternate method of authenticating to the TOE in the event the main passphrase or method of accessing the protected disk is lost. Following recovery passphrase use, a new recovery passphrase is generated by the TOE without user intervention. During each disk encryption operation, the PGP SDK subsystem creates a recovery passphrase for use in the event the authorized user forgets the passphrase used for whole disk encryption. This results in the session key used to encrypt the disk being encrypted with both the user provided passphrase and the Whole Disk Recovery Token created by the PGP SDK subsystem. The recovery passphrase is passed to the PGP Universal Server in the Operational Environment by the ClientLib module (within the WDE Engine) for storage and may be accessed by the PGP Universal Server administrator to assist an authorized user who has forgotten the passphrase.

When the recovery passphrase is used by the authorized user to startup the TSF protected platform, the Bootguard subsystem detects that the recovery passphrase has been used and writes an instruction to the Disk Filter driver that the recovery passphrase has been used and, upon startup, the PGP SDK subsystem will need to create a new recovery passphrase. The PGP SDK subsystem receives the instruction from the Disk Filter driver during the post-boot process, and creates a new recovery passphrase. The new recovery passphrase is then passed to the PGP Universal Server in the Operational Environment by the ClientLib module (within the WDE Engine) for storage. The PGP SDK then re-encrypts the session key with the original User created passphrase and the new recovery passphrase and saves the result to the disk drive. This results in the “single use” mechanism for the recovery passphrase, as the original recovery passphrase can no longer be used to decrypt the session key. In order for the WDRT to regenerate and the used WDRT to be deactivated, use of the WDRT must be followed by an uninterrupted boot and successful normal mode OS Login sequence.

SSLv3.1/TLSv1.0 Sessions between the TOE and Universal Server/Equivalent – FIA_UAU.5

A trusted server certificate must be presented to the TOE by the Universal Server and validated by the TOE prior to allowing the establishment of SSLv3.1/TLSv1.0 sessions. The certificate is validated by verifying the CA is trusted by Windows (in Windows root store) and is therefore considered trusted.

7.1.3 Cryptographic Operations

The TOE utilizes symmetric key cryptography for the purpose of securing the physical disk drive (whole disk encryption) or a Partition(s) within the disk (partition encryption) on the platform. The use of cryptography for this purpose is the primary approach taken by the TOE in securing stored data.

Three keys are used during the whole disk encryption process: A Disk Access Key, a Link Key and a (Disk) Session Key. The Disk Access key is encrypted/decrypted with the passphrase or public key (smartcard or token) used to access the encrypted disk. The Disk Access Key is used as a password to encrypt/decrypt the Link Key. The Link Key is used as a password to encrypt/decrypt the (Disk) Session Key which ultimately encrypts/decrypt the disk blocks of the physical disk drive using 256 bit AES (alg Cert. #1253). In addition to the encryption operations on the keys above, they are also encoded prior to encryption. This provides an additional verification step post decryption. Following decryption of the Disk Access key using the entered passphrase, the TOE verifies that it is encoded correctly prior to proceeding.

The mWDE utilizes the following cryptographic algorithms for the purpose of securing SSLv3.1/TLSv1.0 based communications with the PGP Universal Server in the Operational Environment: AES algorithm (Cert. #954) (128 bit key size), 3DES (Cert. #754) algorithm (192 bit key size). Key exchange is orchestrated between the Universal Server and the TOE using Diffie-Hellman or RSA.

The TOE utilizes two FIPS 140-2, Level 1, validated, software based cryptographic modules: the PGP SDK cryptographic subsystem (FIPS 140-2 Cert. #1101), and the PGP Cryptographic Engine (within the Disk Filter). The applicable FIPS security policies can be located here:

SDK: http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1101.pdf

PGP Cryptographic Engine 4.0: Pending FIPS 140-2 Validation

Cryptographic Key Generation – FCS_CKM.1a, b, c

The mWDE TOE generates symmetric keys using a software RNG for use in Whole Disk Encryption, and Partition Encryption operations. All symmetric key generation for the TOE is performed by the PGP SDK FIPS 140-2 validated cryptographic module. The TOE enforces only the use of secure cryptographic security attributes by default.

For Whole Disk Encryption purposes, the TOE generates 256 bit AES symmetric keys using the 3DES algorithm. The TOE generates symmetric keys using a software-based Random Number Generator (RNG) within the “FIPS 140-2 validated cryptographic module” contained in the PGP SDK subsystem as described in Section 1.7.1.3. Encryption operations use the AES or 3DES algorithm. AES keys are used exclusively for Whole Disk Encryption operations (FCS_CKM.1a), whereas AES or 3DES keys may be used for SSLv3.1/TLSv1.0 sessions with the PGP Universal Server in the Operational Environment.

Session keys are used for Whole Disk or Partition encryption, Link Keys are used to link disks on the same platform together and Disk Access Keys are generated in the background by the PGP SDK cryptographic subsystem at the time of the selected encryption activity by entering the applicable passphrase (Passphrase Authentication), using a passphrase through a supported TPM or by specifying a supported public key within a supported Smartcard or token device. The authorized user may also select to use the Windows password for encryption, thereby allowing for a single sign-on (passphrase-single sign-on authentication).

Passphrases are converted to key material through a salted “String to Key” (S2K) operation performed at different times by both the PGP SDK subsystem and Bootguard module. The passphrase represents a string that is hashed multiple times using SHA1 with a 16 bit salt to result in a key that is used in a secondary operation to symmetrically encrypt the Disk Access Key. When the Passphrase is originally established to protect a disk resource, the S2K operation is performed by the PGP SDK module. During startup operations, the passphrase entered to access the disk is hashed/salted by the Bootguard subsystem using the same technique. This S2K operation conforms to OpenPGP RFC 4880.

Regardless of the method of invocation, keys used for whole disk or partition encryption are generated in the same manner by the PGP SDK cryptographic subsystem of the TOE. Additional details regarding Key Generation are provided below under Cryptographic Key Access – FCS_CKM.3.

TSF encryption operations of the physical disk drive exclusively use the AES (Cert. #954) algorithm and 256 bit key size operating in CFB block cipher mode. This symmetric cipher selection is hard coded and cannot be changed by the user of the application.

Key generation operations are invoked through the PGP SDK API by the WDE Engine subsystem when part of a Whole Disk/Partition encryption operation.

Cryptographic Key Access – FCS_CKM.3, FCS_CKM.1c

Encrypted Disk access during the boot and post boot processes is controlled by the entry of a User Passphrase, presentation of a Smartcard or token (public key), or use of a Trusted Platform Module (passphrase).

The passphrase (or token key) must be presented to ultimately access the session key required to decrypt a given hard drive resource.

Protection of the (Disk) Session Key is accomplished through the following steps based on the access method:

User Passphrase – symmetric key

The User Passphrase entered at the time of Disk Encryption is iterated using the "iterated salted string to key method" described in OpenPGP RFC 4880. This technique is used with a 16 byte salt vs. the 8 byte salt described in the RFC. This provides the applicable key encryption key. The payload is then padded up to 128 bytes using OAEP padding (PKCS#1) and is encrypted using AES 256 (Cert. #954) (CBC mode) and the encryption key mentioned above.

Note: This same technique is used to encrypt the 3 types of symmetric keys used for disk encryption operations: the Disk Access Key (unique to the disk group), the Link Key (links disks on the same computer) and the (Disk) Session Keys which encrypt the disk blocks themselves. By the user entering the user passphrase to decrypt the Disk Access key, the Link Key is decrypted and then is used to decrypt the Session Key, which ultimately allows access to the encrypted disk.

Smartcard or token based authentication - (public key)

The first step in smartcard or token based authentication for disk encryption is for the user to authenticate to their token device. The PIN is entered through software or directly on the hardware PIN pad device (these Token devices are part of the Operational Environment). The payload is padded to 128 or 256 bytes using PKCS#1 OAEP or PKCS#1 legacy padding. The padding size is based on whether 1024 or 2048 bit RSA public key is used. The payload is then encrypted using the public key on the token. The encrypted Disk Access key is then stored on the smartcard or token and the smartcard or token performs an RSA decryption operation and passes the Disk Access Key to the TOE application to result in access to the encrypted disk. The Smartcard or token itself is identified by the TOE through querying during the Pre-boot process or via software modules such as PKCS#11 during the OS session. These operations are performed by the applicable Smartcard or Token; the TOE simply provides an interface to these devices.

Trusted Platform Module (TPM) authentication – passphrase

The payload is passed to the TPM along with a SHA-1 hashed password used to seal the data. The sealing is done using the Storage Root Key (SRK) of the TPM. The TPM SRK is typically a 2048 bit RSA key. The sealing operation requires that the user provided, hashed password is correct in order to decrypt the payload using the SRK of the TPM. This is the aspect that binds the encryption operation to the TPM. The sealed Disk Access Key is stored in the TPM user record on the physical disk. Upon proper authentication the TPM performs the "unseal" operation and passes the Disk Access Key to the TOE application to result in access to the encrypted disk. The Trusted Platform Module itself is identified by the TOE through querying during the Pre-boot process or via software modules such as CAPI during the OS session.

In all cases, the Disk Access Key is decrypted based on either a User provided passphrase or public key (smartcard or token) that ultimately allows access to the encrypted physical disk.

The Disk Access is never cached for persistent access. Once the Disk Access key is acquired through the methods described above, it may in turn decrypt the Link Key, which in turn decrypts the (Disk) Session key which ultimately is cached by the TOE to allow protected data to be decrypted as accessed by the user through the Disk Filter subsystem.

Post-boot Cryptography key access for application management:

The TOE requires that credentials be entered to release associated cryptographic keys for a limited set of security management operations including: disk decryption, re-encryption of a disk resource, adding or removing users, and changing the user passphrase.

Cryptographic Key Destruction – FCS_CKM.4

Cryptographic keys are destroyed by the PGP SDK subsystem through zeroization (single pass overwrite with random data) of the volatile memory location caching the key material. Key material is maintained in non-paged memory locations and is not written to underlying OS swap files.

The cryptographic key formed by the hashing of the passphrase during login is destroyed through zeroization as soon as the session key is released into cache, allowing the boot process to proceed. The only exception to this is in the case of the Windows Single sign-on option, where the passphrase is cached until the Windows login is completed and is then zeroized.

Cryptographic Operations, Whole Disk Encryption, Partition Encryption – FCS_COP.1a

Whole Disk Encryption and Partition Encryption is executed on a target resource by the authorized user selecting the whole disk or partition encryption option from the PGP Desktop GUI interface. Upon selecting the target for encryption, the authorized user selects the method for authenticating the user to the resource following encryption using either a passphrase based method, Windows single sign-on, Smart Card or Trusted Platform module. Based on the selection, the session keys are generated by the PGP SDK subsystem and the PGP Cryptographic Engine within the Disk Filter performs whole disk encryption on the target disk or partition. Following the encryption operation, the session key used for encryption is itself encrypted using the acquired key material provided by the selected authentication method. The result is a symmetric session key encrypted by the hashed passphrase. This encrypted session key is then stored on the disk drive.

Whole Disk/Partition Encryption-Decryption Operations – FCS_COP.1a

The WDE Engine subsystem orchestrates encryption/decryption operations while running in user mode on the installed platform. The Disk Filter subsystem using the PGP Cryptographic Engine supports disk read/write operations in association with the WDE Engine and runs in the Operating System kernel space. The PGP Cryptographic Engine leverages an AES implementation to perform encryption/decryption on data as it is access from the encrypted disk resource. The Disk Filter and WDE Engine communicate through Input/Output (I/O)

Control; when the WDE Engine requires a disk operation it creates a custom I/O Control call to signal an event that the WDE Engine is listening for, to the Disk Filter driver which executes the applicable action.

When the Whole Disk Encryption feature is first implemented, the original Master Boot Record is replaced with the PGP MBR (subsystem) object. The Bootguard file system (Bootguard subsystem) is installed on the physical disk along with details regarding the status of the disk and user information. This information is accessed by the BIOS Filter during the boot process to ascertain encryption status of the disk, such as authentication method in use, without user intervention.

The physical disk is encrypted by the authorized user first adding or selecting a user account and specifying an authentication method to be used for disk access once encrypted. The WDE Engine calls the PGP SDK cryptographic subsystem which generates the disk session key and contacts the Disk Filter driver with the key information so the encryption operation can proceed. The PGP Cryptographic Engine begins encrypting the drive one block at a time from sector 0 to the last sector. For example, if a disk has 2560 sectors, the Disk Filter will encrypt 256 sectors in a single transaction. It reads the first 256 sectors, encrypts them with the session key and then writes the encrypted content to the location of the first 256 sectors. In this example, it takes 10 transactions to complete the initial disk encryption process. Following the initial encryption process, disk encryption/decryption activities occur on the fly by the AES implementation within the PGP Cryptographic Engine.

During the disk encryption process, normal disk I/O transactions are still allowed to occur. In the event the Operating System component attempts to read a sector that has already been encrypted, the Disk Filter driver will receive the request and read the requested sector into a buffer supplied by the original reading thread. The Disk Filter driver decrypts the sector and notifies the reading thread that the required I/O is complete.

If an attempt is made to write to a sector during the disk encryption process that has already been encrypted, the writing thread supplies a buffer to the Disk Filter driver that contains the content for the targeted sector. The Disk Filter driver then encrypts this data and writes it to the applicable thread. Plaintext cannot be written to the drive at any time once encrypted by the TSF.

Once the Whole Disk Encryption is complete and on all subsequent startup cycles, the Disk Filter operates as a proxy and decrypts during read requests and encrypts during write requests.

Key usage during Whole Disk Encryption/Decryption operations

The (Disk) Session Key used to encrypt the Physical disk during disk encryption operations is an AES, 256 bit symmetric key. The usage of the AES algorithm and 256 bit key size is hard coded into the application and cannot be changed by any user. This key is encrypted with the Link Key generated by the PGP SDK cryptographic subsystem which itself is encrypted with the applicable Disk Access key encrypted using a hash of the authorized user's authentication data (passphrase or smartcard or token key) and is stored on the disk drive. The release of the Disk Access Key through the entry of the correct passphrase or presentation of the applicable Token public key ultimately results in the Session Key being read into cache where it is used to decrypt disk data.

This Session Key once released following authentication is sent to the Disk Filter Driver and is stored in kernel memory. The key remains in this memory location until the platform is rebooted and the Bootguard subsystem again presents the authentication login dialog. Any kernel or user mode component indirectly utilizes the disk session key when access disk content as the disk filter driver "proxies" all disk I/O operations.

Following the pre-boot authentication process, access to the disk is granted and the TOE provides no further authentication requirements or access restrictions at the disk level.

Administrator Session Encryption, TOE application to PGP Universal Server – FCS_COP.1b, FCS_COP.1c

The TOE establishes secure sessions with the PGP Universal Server (or equivalent) in the Operational Environment for the purpose of downloading configuration settings relating to mWDE local policies and for uploading audit logs. This data is uploaded/downloaded to a specific file location on the PGP Universal Server, associated with a particular authorized user/mWDE installation. This server cannot directly access the TOE application directly in any manner and the TOE application (Client) can only access the allocated file location on the PGP Universal Server for purposes of uploading audit log data or downloading policy related security attributes.

These sessions are secured via SSLv3.1 or TLSv1.0 protocols using the AES (Cert. #954) or 3DES (Cert. #754) symmetric algorithm with a SHA-1 hash. The sessions are initiated by the TOE application (client) through a handshake by the PGP Universal Server and a Diffie-Hellman key exchange is executed to establish a shared secret and generate a SSLv3.1/TLSv1.0 session cryptographic key. Cryptographic operations and the TLSv1.0 implementation used for securing mWDE to PGP Universal session is provided by the PGP SDK (FIPS 140-2 validated) subsystem. The SSLv3.1/TLSv1.0 implementation within the PGP SDK Cryptographic subsystem manages all facets of the Diffie-Hellman key agreement protocol and SSLv3.1/TLSv1.0 implementation necessary to establish the secure session between the TOE application and the PGP Universal Server.

The SSLv3.1/TLSv1.0 implementation contained within the PGP SDK Cryptographic subsystem is proprietary to PGP Corporation.

The communication between the PGP Universal Server and the TOE (client) is protected by server-side authentication via SSLv3.1/TLSv1.0. As configured to Common Criteria evaluation guidance, the server holds a certificate that is passed to the TOE and recognized as a trusted certificate. Within this protected channel the TOE provides a "cookie" (some random data) to the PGP Universal Server that was originally passed to the TOE by the PGP Universal Server during initial configuration. This cookie is protected by Microsoft APIs that limit its access to only the original logged on user who enrolled. This certificate exchange is required to initiate the Diffie-Hellman key exchange.

7.1.4 Security Management

The TOE provides a comprehensive Security Management interface that allows the local user to generate and manage cryptographic keys, execute encryption and decryption operations against physical disk drive resources, configure application options and view audit logs. The Security Management security function is supported by the User Interface subsystem and the WDE Engine subsystem.

PGP® mWDE Security Management Functions – FMT_SMF.1

The TOE GUI based security management interface is divided into two major categories of management functions, “PGP Keys” which includes all the key generation and key management functions and “PGP Disk”, which presents the options for encrypting the Whole Disk or Partition Encrypt.

GUI Interface

The GUI Interface is provided by the User Interface subsystem. The UI is a standard Win32 application which is rendered using win32 UI calls; that is it relies on the Windows UI subsystem. Options made in the UI are stored either in the PGP preferences file (an XML file that is stored in the user directory) or within the TSF data that is stored on the disk. Disks can be encrypted or decrypted through this interface and essential user management functions allow the creation of accounts and updating of passphrases. Despite the fact that multiple accounts may be created through this interface, the CC Evaluated configuration stipulates that no more than a single user per platform is maintained.

Security attributes managed through this interface include authentication related attributes and key related attributes including:

> Authentication attributes: Passphrases, Smartcard or token usage assignment, Trusted Platform Module assignments.

These attributes are passed from the UI subsystem (GUI) to the WDE Engine and then to the PGP SDK cryptographic subsystem for implementation. These attributes are used by the Bootguard subsystem, the Disk Filter subsystem and the Bootguard subsystem during the TOE authentication process.

Audit logs - FMT_SMF.1

Audits logs are accessed through the security management interface via a pop-up window that launches when the "view log" option is selected from the pull down menu. The log info is stored in a simple RTF (rich-text format) file and the UI displays the contents of the file in a MS RTF widget. When filters are applied, the UI parses the log file and only displays those lines in the file that match the filter. The audit logs may be filtered by the following logging levels Verbose, Info, Warning, Error and by the following application related categories All, PGP, Email, IM, Whole Disk. A date range filter can also be applied that includes date ranges including: "today", "yesterday", 2 – 7 days ago.

Cryptographic functions configuration & management – FMT_SMF.1

Key management within the TOE application is managed through a series of GUI management screen in the PGP Keys category. The local user can generate public/private keys pairs, change passphrases associated with key pairs, enable/disable keys for use, sign keys, revoke keys and export keys to an ASCII file format through this interface.

When a WDE or Partition encryption operation is selected, this interface may also be used to create a User Account within the application and allow the creation of a passphrase or assignment of a smartcard or token keypair to be used to authentication purposes.

When key changes are implemented, the UI subsystem passes the information associated with the change to the WDE engine which accesses the PGP SDK cryptographic subsystem to enact the requested change.

TSF Data imported by the TOE from PGP Universal Server

The TOE imports policies through XML based commands which are downloaded through secure sessions with the Universal Server. The local authorized mWDE (TOE) user cannot access these policies in any form on the local platform. In addition, only the Universal Server Administrator in the Operational Environment may configure the authentication failure threshold that applies to the TOE via polices established on the Universal Server and imported by the TOE.

Management of Security Function Behavior – FMT_MOF.1a, b

The Security Management interface provides the primary vehicle for managing the behavior of security functions by allowing the authorized user to specify which resources are to be encrypted/decrypted and which smartcard or token keys/passphrase to use.

The selection or de-selection of these options effects security function behavior through the UI subsystem passing the selection attribute to the WDE Engine and then to applicable subsystems where it is implemented and saved to a configuration file on the local drive within the application folders.

The TOE requires the entry of authentication credentials prior to allowing encryption or decryption operations to be initiated by the TSF.

Security function behavior may also be managed via policies created on the PGP Universal

Server in the Operational Environment and imported by the TOE. These policies may enable or disable specific functions that will include/remove those functions from the PGP mWDE GUI interface. Security functions that can be enabled or disabled in this manner include: Whole Disk Encryption or Decryption, Automatic encryption of the boot disk, Require a particular authentication method (i.e. TPM), force encryption of removable USB drives, enable/disable WDE recovery tokens, Allow/Disallow Windows single sign-on. As intended, Policies imported by a local mWDE implementation may override or otherwise restrict the local user's ability to implement mWDE functions.

The security management attributes downloaded from the PGP Universal Server by the TOE application are passed along with the rest of the internal user policy to the client when it requests policy from the server. They are downloaded over an SSLv3.1 encrypted SOAP connection. They are represented as parts of the XML that represent the overall policy. The client then makes that XML a sub-piece of a larger XML document that represents its local policy and the server policy.

These attributes are accessed by the TOE during the startup process, where the ClientLib module within the WDE Engine accesses an account within the PGP Universal Server to access policy related attributes as noted above.

TSF Data Management – FMT_MTD.1

TSF data managed by the TOE includes key material and associated passphrases, user identification such as user name and email address; audit logs maintained by the TOE, and passphrases generated by the TOE (recovery tokens).

Access to TOE management screens which manage TSF data is allowed for any post boot user, however, the correct credentials are required to be entered for a local user to create, modify or delete passphrases associated with disk or partition encryption keys and adding, removing or modifying user TOE accounts. The entered credentials release cryptographic keys associated with these operations.

7.1.5 Protection of the TOE

PGP® mWDE protection – FCS_COP.1a, FCS_COP.1b, FPT_ITI.1

The TOE application protects the underlying application and Operating System TOE component by employing the WDE encryption function. By fully encrypting the physical drive and protecting the disk BIOS through the Bootguard subsystem, potential attackers are unable to access platform hard drive resources. In addition, the passphrase or smartcard or token key used to access the drive is never saved on the hard disk in its unencrypted form. The passphrase/smartcard or token pair is used to encrypt the WDE symmetric encryption key which is then stored on the disk drive media.

All cryptographic key material when stored on the hard disk is encrypted using AES, 256 bit keys. Keys reside in plaintext only in volatile, non-paged memory location where it is zeroized following use or platform reboot/shutdown.

The TOE's drivers, executables and dynamic link libraries (.dll) are all signed with a VeriSign® code-signing certificate to verify authenticity and assure object integrity. These are verified during installation to assure that the TOE software is not compromised prior to installation.

TSF access is unrestricted Post-Boot except for the following functions that require users to enter the applicable credentials to release required cryptographic keys: disk decryption, re-encryption of a disk resource, adding or removing users, and changing the user passphrase. Therefore, the primary TOE protection is afforded through encryption of the platform drive resource and pre-boot authentication requirements.

The TOE application protects data transfers to and from the PGP Universal Server by securing sessions using either SSLv3.1 or TLSv1.0 using FIPS 140-2 validated cryptography. By encrypting these sessions, and through utilization of the SSLv3.1/TLSv1.0 protocol, malicious users are prevented access to the application through this interface and data transfer modifications during sessions are detected by the TOE.

7.2 Rationale for TOE Security Functions

This section provides a table demonstrating the tracing of TOE security functions back to aspects of the security functional requirements (SFRs). A justification that the security functions are suitable to cover the SFRs can be found in Section 7.1.

	Security Audit	Identification and Authentication	Cryptographic Operations	Security Management	Protection of the TOE
FAU_GEN_EXP.1	X				
FCS_COP_EXP.1		X			
FAU_SAR.1	X				
FAU_SAR.3	X				
FCS_CKM.1a			X		
FCS_CKM.1b			X		
FCS_CKM.1c			X		
FCS_CKM.3			X		
FCS_CKM.4			X		
FCS_COP.1a			X		X
FCS_COP.1b			X		
FCS_COP.1c			X		

FIA_AFL.1		X			
FIA_UID.2		X			
FIA_UAU.2		X			
FIA_UAU.5		X			
FIA_UAU.7		X			
FMT_MOF.1a,b				X	
FMT_MTD.1				X	
FMT_SMF.1				X	
FPT_ITI.1					X

Table 14: TOE Security Function to SFR Mapping