Security Target

---

# McAfee File and Removable Media Protection 4.3.1

# and

# ePolicy Orchestrator 5.1.2

Document Version 1.0

6 september, 2015

*Prepared For:*

**Intel Corporation**

**2821 Mission College Blvd.**

**Santa Clara, CA 95054**

*Prepared By:*

**Primasec Limited**

**Le Domaine de Loustalviel**

**11420 Pech Luna, France**

## Abstract

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE): McAfee File and Removable Media Encryption 4.3.1 and ePolicy Orchestrator 5.1.2. This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements, and a specification for the IT security functions provided by the TOE that meet the set of requirements.

# Table of Contents

## List of Tables

# List of Figures

# 1 Introduction

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

## 1.1 ST reference

| | |
|---|---|
| **ST Title** | Security Target: McAfee File and Removable Media Protection 4.3.1 and ePolicy Orchestrator 5.1.2 |
| **ST Revision** | 1.0 |
| **ST Publication Date** | 6 September 2015 |
| **Authors** | Primasec Limited and Intel Corporation |

## 1.2 TOE reference

| | |
|---|---|
| **TOE Reference** | McAfee File and Removable Media Protection 4.3.1 and ePolicy Orchestrator 5.1.2 |
| **TOE Type** | Encryption software |

## 1.3 Document organization

This Security Target follows the following format:

| SECTION | TITLE | DESCRIPTION |
|---|---|---|
| 1 | Introduction | Provides an overview of the TOE and defines the hardware and software that make up the TOE as well as the physical and logical boundaries of the TOE |
| 2 | Conformance Claims | Lists evaluation conformance to Common Criteria versions, Protection Profiles, or Packages where applicable |
| 3 | Security Problem Definition | Specifies the threats, assumptions and organisational security policies that affect the TOE |
| 4 | Security Objectives | Defines the security objectives for the TOE/operational environment and provides a rationale to demonstrate that the security objectives address the threats |
| 5 | Extended Components Definition | Describes extended components of the evaluation |

| SECTION | TITLE | DESCRIPTION |
|---|---|---|
| 6 | Security Requirements | Contains the functional and assurance requirements for the TOE |
| 7 | TOE Summary Specification | Identifies the IT security functions provided by the TOE and also identifies the assurance measures targeted to meet the assurance requirements. |

**Table 1 – ST Organization and Section Descriptions**

## 1.4 Document conventions

The notation, formatting, and conventions used in this security target are consistent with those used in Version 3.1 of the Common Criteria. Selected presentation choices are discussed here to aid the security target reader. The Common Criteria allows several operations to be performed on functional requirements: the allowable operations defined in Part 2 of the Common Criteria are *refinement, selection, assignment* and *iteration*.

- The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. An assignment operation is indicated by *italicized* text, contained within square brackets.
- The refinement operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by **bold text**. Any text removed is indicated with a strikethrough format (Example: ~~TSF~~).
- The selection operation is picking one or more items from a list in order to narrow the scope of a component element. Selections are denoted by underlined text, contained within square brackets.
- Iterated functional and assurance requirements are given unique identifiers by appending to the base requirement identifier from the Common Criteria an iteration number inside parenthesis, for example, FIA_UAU.1 (1) and FIA_UAU.1 (2) refer to separate instances of the FIA_UAU.1 security functional requirement component.

Outside the SFRs, italicized text is used for both official document titles and text meant to be emphasized more than plain text.

## 1.5 Document terminology

The following tables describe the terms and acronyms used in this document:

| TERM | DEFINITION |
|---|---|
| API | Application Programming Interface |
| CC | Common Criteria |
| CM | Configuration Management |

| TERM | DEFINITION |
|---|---|
| EAL | Evaluation Assurance Level |
| ePO | ePolicy Orchestrator |
| GB | Giga-Byte |
| GUI | Graphical User Interface |
| GUID | Globally Unique identifier |
| I&A | Identification and Authentication |
| IT | Information Technology |
| MA | McAfee Agent |
| MB | Mega-Byte |
| MCCM | McAfee Common Cryptographic Module |
| MFRP | McAfee File and Removable media Protection |
| MFS | McAfee Foundation Service |
| OS | Operating System |
| OSP | Organisational Security Policy |
| PC | Personal Computer |
| PBKDF | Power-Based Key Derivation Function |
| PP | Protection Profile |
| RAM | Random Access Memory |
| SFR | Security Functional Requirement |
| SMTP | Simple Mail Transfer Protocol |
| SNMP | Simple Network Management Protocol |
| ST | Security Target |
| TOE | Target of Evaluation |
| TSC | TOE Scope of Control |
| TSF | TOE Security Function |
| TSFI | TOE Security Function Interface |

**Table 2 –Acronyms Used in Security Target**

| Term | Definition |
|---|---|
| Authorized administrator | An ePO user assigned the appropriate permission for the operation being performed or an ePO administrator |

**Table 3 –Terms Used in Security Target**

## 1.6 TOE overview

McAfee File and Removable media Protection (FRP) allows selective control of access to data held in file systems and on removable media, based on user permissions. This protection depends on Microsoft Windows user accounts and works in real-time to authenticate the user, to access the encryption keys, and to retrieve the correct policy in FRP. A smart card implementation based on Windows logon provides for enhanced security.

FRP encrypts files and folders as per the policies assigned to users. These policies are enforced by the McAfee ePO server.

FRP acts as a Persistent Encryption engine. When a file is encrypted and is moved or copied to another location, it remains encrypted. If it is moved out of an encrypted directory, it still remains encrypted.

Integrated with McAfee® ePolicy Orchestrator® (McAfee ePO™), FRP provides a single point of control over the data on all systems, and supports both user and system based policies. FRP depends on Microsoft Windows credentials, thus both registered domain users and local system users can be assigned encryption policies and associated keys. Assigning these policies to users encrypts the data on the client. User-based policy assignments can be assigned only to registered domain users.

The FRP client is installed on the managed system, and then the system synchronizes with the McAfee ePO server and acquires the user data. FRP then assigns encryption policies and keys to the user.

FRP client acts like a filter between the application creating or editing the files and the storage media. When a file is saved, the FRP filter executes the assigned encryption policies and encrypts the data, if applicable.

## 1.7 TOE Description

These are the key features of FRP with ePO.

- **Centralized management** — Provides support for deploying and managing FRP using McAfee ePO software 5.1.
- **Windows authentication based policy enforcement** — Assigns encryption policies and keys to Windows user accounts.
- **Integration with the McAfee tray icon** — Consolidates the tray icons into one common McAfee icon.
- **User Personal Key** — Allows users to have individual encryption keys that are generated from the McAfee ePO server, which the ePO administrators and ePO users can assign to policies to enable encryption.
- **Protect data on removable media** — Removable media encryption, including the ability to access FRP encrypted content in systems where FRP is not installed.
- **Network encryption** — Enables secure sharing and collaboration on Network Shares.
- **User initiated encryption of files and email attachments** — Allows users to create and attach password encrypted executable files that can be decrypted on systems where EEFF is not installed.

The TOE consists of three components: FRP, ePO and McAfee Agent.

### 1.7.1 ePolicy Orchestrator (ePO)

ePO distributes and manages agents that reside on client systems. ePO provides the central management interface and functionality for the ePO administrators and ePO users of the TOE. It also provides key storage, reporting and product deployment capabilities, all through a single point of control. An extension to ePO provides FRP specific management features.

### 1.7.2 FRP

FRP is the client portion of the TOE, and runs on PC workstations. It provides the encryption and decryption service, implementing policies downloaded from ePO appropriate to the system and logged in workstation user. Encryption/decryption operations are normally done using keys downloaded from ePO. A policy setting within ePO allows users to create local keys on a client system.

### 1.7.3 McAfee Agent

McAfee Agent is a vehicle of information and enforcement between the ePO server and each managed system. It provides common communication functionality between ePO and all of McAfee's product-specific software (such as FRP).

### 1.7.4 Encryption keys

FRP uses encryption keys to protect files and folders on networks, removable media, CDs or DVDs, and user hard disks. Encryption keys are normally generated and stored in an internal key repository within the McAfee ePO environment.

The FRP client requests an encryption key when a user accesses an encrypted file or a folder. If a policy is assigned to the user with the requested encryption key, FRP decrypts the data. An ePO administrator can create and manage encryption keys from McAfee ePO. These keys are assigned to policies that are later assigned to users or systems. All keys assigned through the policy are loaded at every log on.

FRP supports three types of encryption keys:

- **Regular keys** — Created by McAfee ePO administrators and can be used in any policy.
- **User personal keys** — Generated in McAfee ePO when a key is granted to a user through the Grant Key policy. When assigned to a user, these policies enable the user to use the key across all the client systems in the same domain.
- **User Local keys** — Created using FRP client software on a client system. These keys can be used by the user to encrypt or decrypt data on the same network using the context menu. A Local key is normally limited to the user and client system on which it was created, although such keys can be exported from one system and imported to another for use in encryption and decryption.

### 1.7.5 Physical boundary

The TOE is a software TOE and includes:

1. The ePO application executing on a dedicated server;
2. The McAfee Agent and FRP software on each client to be protected.

The physical components of the TOE include the software that is installed during installation of FRP, McAfee Agent and ePO. The TOE software is installed on a centralized ePO server and on client workstations. The computer hardware platforms that the TOE software is installed on are not part of the TOE.

The components of the TOE are installed on systems with resident operating systems, but the operating systems are not part of the TOE.

ePO requires a database, but the DBMS is not part of the TOE.

The following documentation provided to end users is included in the TOE boundary:

1. *McAfee File and Removable Media Protection 4.3.0 Product Guide*[1]
2. *McAfee File and Removable Media Protection 4.3.0 User Guide*
3. *McAfee ePolicy Orchestrator 5.1.0 Software Product Guide*[2]
4. *McAfee ePolicy Orchestrator 5.1.0 Software Installation Guide*
5. *McAfee Agent 5.0.0 Product Guide*
6. *McAfee File and Removable Media Protection 4.3.1 with ePolicy Orchestrator 5.1.2 Common Criteria Evaluated Configuration Guide*

In order to comply with the evaluated configuration, the following hardware and software components should be used:

| TOE COMPONENT | VERSION/MODEL NUMBER |
|---|---|
| TOE Software | FRP 4.3.1.114<br>ePolicy Orchestrator 5.1.2.348<br>McAfee Agent 5.0.0.2620 |
| IT Environment | Specified in the following:<br>• Table 5 – Management System Component Requirements<br>• Table 6 – Managed System Platforms |

**Table 4 – Evaluated Configuration for the TOE**

The evaluated configuration includes one or more instances of McAfee Agent and FRP and an instance of ePO.

The following figure presents an example of an operational configuration. The shaded elements in the boxes represent the TOE components.

[1] Guides for 4.3.0 are also applicable to 4.3.1.
[2] Guides for 5.1.0 are also applicable to 5.1.2.

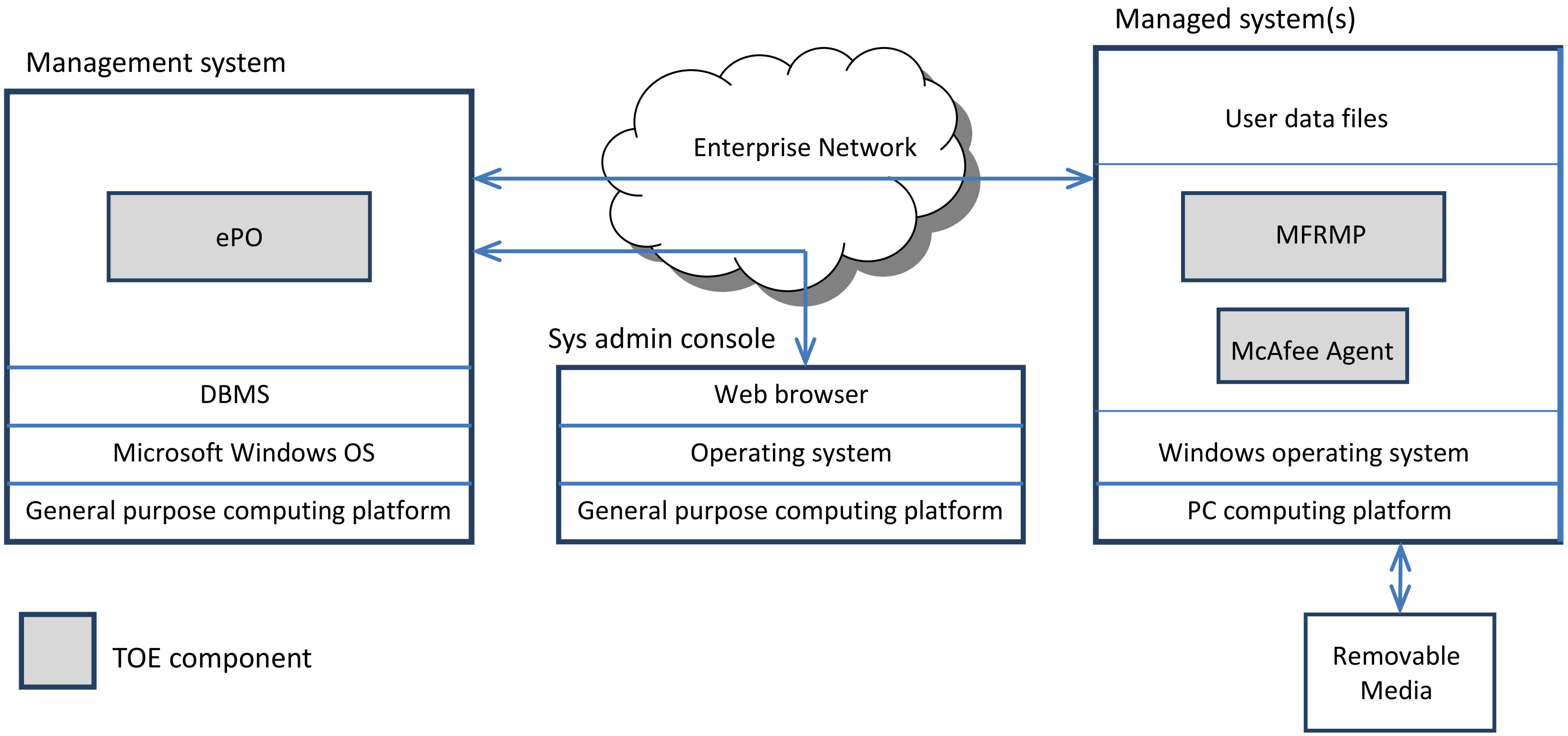


**Figure 1 – TOE boundary**

The following are the main components of the overall solution:

- MA: The McAfee Agent (MA) is software that resides on the workstation along with the FRP software and is responsible for communicating with the ePO server.
- ePO Console: This is part of the ePO server infrastructure in the enterprise network through which the IT organization will deploy and manage the McAfee security solution on the devices.
- FRP client: The client software that performs encryption/decryption operations in line with policy.
- DBMS: (Operational Environment) The database stores ePO user accounts, permissions, permission sets, assets, policies, policy templates, and events.

### 1.7.6 Hardware and software supplied by the IT environment

The TOE consists of a set of software applications. The hardware, operating systems and all third party support software (e.g. ePO DBMS) on the systems on which the TOE executes are excluded from the TOE boundary.

The platform on which ePO is installed must be dedicated to functioning as the management system. ePO operates as a distribution system and management system for a client-server architecture offering components for the server part of the architecture (not the clients). The TOE requires the following hardware and software configuration on this platform.

| COMPONENT | MINIMUM REQUIREMENTS |
|---|---|
| Processor | Intel Pentium 4-class or higher; 2.66 GHz or higher |
| Memory | 8 GB RAM |
| Free Disk Space | 2.5 GB |
| Monitor | 1024x768, 256-color, VGA monitor or higher |
| Operating System | Windows Server 2008 R2 |
| Additional Software | Microsoft SQL Server 2008 R2, 2012<br>LDAP server (to enable dynamically assigned permission sets)<br>Web browser |
| Network Card | Ethernet, 100Mb or higher |
| Disk Partition Formats | NTFS |
| Domain Controllers | The system must have a trust relationship with the Primary Domain Controller (PDC) on the network |

**Table 5 – (ePO) Management System Component Requirements**

The supported platforms for McAfee Agent and FRP are:

| COMPONENT | MINIMUM REQUIREMENTS |
|---|---|
| Processor | 1Ghz or faster |
| Memory | 1GB (32-bit), 2GB (64-bit) |
| Free Disk Space | 50 MB |

| COMPONENT | MINIMUM REQUIREMENTS |
|---|---|
| Operating System | Microsoft Windows  8.1(32-bit and 64-bit)<br>Microsoft Windows 7 (Professional, Enterprise or Ultimate) SP1 (32-bit and 64-bit) |
| Network | TCP/IP network connection |

**Table 6 – Managed System (FRP Client) Platforms**

The (ePO) management system is accessed from remote systems via a browser.

Identification and authentication services for ePO users are provided by either the TOE (in the case of local ePO password authentication) or by the Windows operational environment (in the case of Windows authentication), and for workstation users by the Windows operational environment.

Protection of communications between the FRP client and ePO is handled by McAfee s-pipe, and that between ePO and a remote management browser by HTTPS in the TOE environment.

### 1.7.7 Logical boundary

The logical boundaries of the TOE are defined by the functions provided by the TOE and are described in the following sections.

| TSF | DESCRIPTION |
|---|---|
| Cryptographic support | ePO generates and stores cryptographic keys that are deployed to encrypt and decrypt protected user data. Where permitted by policy, FRP also generates and stores local keys for protection of data under user control. Cryptographic protection is also provided for TSF data in transit. |
| User data protection | FRP implements policies defined on ePO to limit access to specified user data through use of encryption. Encryption can be applied to specified pathnames within a file system, or to specified types of file. Protection can also be applied to removable media and to network locations. |
| Management | ePO enables an administrator to centrally manage security settings for the managed workstations. |
| Identification and authentication | FRP Client uses Windows authentication to determine the logged in user, and provides password based features for modules such as User Local Keys, Self-Extractor and features relates to removable media.<br>ePO Server uses either Windows authentication or ePO local password authentication, as determined by the administrator configuration to determine the permission sets assigned to the administrator. |
| Protection of the TSF | Cryptographic key material and policies are passed from ePO to managed workstations, and are protected against disclosure and modification. |

**Table 7 – Logical Boundary Descriptions**

### 1.7.8 TOE data

TOE data consists of both TSF data and user data (information). TSF data consists of authentication data, security attributes, cryptographic keys and other generic configuration information. Security attributes enable the TSF to enforce the security policy. Authentication data enables the TSF to identify and authenticate users (either ePO users/administrators or, in the case of self-extractors, user local keys, CD/DVD/ISO and other removable media, FRP Client users).

| TSF Data | Description | AD | UA | GE |
|---|---|---|---|---|
| Audit Log | Audit events relating to administrator activity on the ePO server. | | | ✓ |
| Client events | Audit events relating to activity on the FRP clients recording enforcement of FRP policies on the managed system | | | ✓ |
| Dashboards | Collections of chart-based queries that are refreshed at a user-configured interval. | | | ✓ |
| ePO User Accounts | ePO user name, authentication configuration, enabled status, and permission sets for each user authorized to access TOE functionality on ePO. | ✓ | ✓ | |
| Permission Set | A group of permissions that can be granted to any user by assigning it to the user’s account. | | ✓ | |
| Queries and reports | Configurable objects that retrieve and display data from the database. | | | ✓ |
| Registered Servers - LDAP | Controls the Active Directory with which ePO synchronizes system and user data | | | ✓ |
| Systems | Information specific to a single managed system (e.g. internet address) in the system tree. | | | ✓ |
| System Tree | A hierarchical collection of all of the systems managed by ePolicy Orchestrator. | | | ✓ |
| FRP Regular Keys | Created by ePO Administrators (and ePO users with FRP Key Server: Manage Key Server permission), and can be used in any policy. Can be assigned at the system level, or to users via a policy assignment rule. | | | ✓ |
| FRP Personal Keys | Keys generated on ePO for encryption of user data, when a key is granted to a user. Stored on ePO and distributed across a domain as required. | | ✓ | |
| FRP User Local Keys | Keys generated locally on the FRP client for encryption of user data | | | ✓ |
| FRP Passwords | Passwords used for self-extractors, user local keys, CD/DVD/ISO and other removable media. | ✓ | | |

**Table 8 – TOE Data**

(Legend: AD=Authentication data; UA=User attribute; GE=Generic Information)

## 1.8 Operating system platforms not covered by the evaluation

The File and Removable Media Protection product can also be installed on the following client platforms, but these are not covered by this evaluation:

- Microsoft Windows 8.1 Update
- Microsoft Windows 8
- Microsoft Windows XP Professional SP3 (32-bit only)
- Microsoft Windows Vista (Business, Enterprise or Ultimate) SP2 (32-bit and 64-bit)
- Apple Mac OS X Mountain Lion 10.8.0 and Mavericks 10.9.0 or above
- Virtual Desktop Infrastructure (VDI) Citrix XenDesktop 5.6 and Citrix XenDesktop 7.1

## 1.9 Rationale for non-bypassability and separation of the TOE

The TOE is an application that executes on top of an underlying hardware system in conjunction with a Windows operating system. Responsibility for non-bypassability and separation are split between the TOE, the OS and the hardware IT Environment.

All access to objects in the TOE IT environment is validated by the IT environment security policies before they can succeed. An attacker will not be able to access any of the TOE security functions or any of the TOE files or directories. Arbitrary entry into the TOE is not possible, and therefore the TSF is protected against external interference by untrusted objects.

The TOE provides strictly controlled functionality to the users within the TSC. By limiting access through ePO role based access control, the TSF is protected from corruption or compromise from users within the TSC. The TOE interfaces are separated into 2 categories – security enforcing and security supporting. Security enforcing interfaces invoke the TSF and ensure that all enforcement functions complete successfully before allowing the user invoked action to proceed. Security supporting interfaces ensure that the TSF cannot be interfered with via those interfaces (i.e. they are isolated from the TSF). The security enforcing role is separate from the security supporting role and each role has its own unique set of privileges associated with it. Multiple simultaneous users (and roles) are supported.

The hardware and OS provide separate process spaces in which the TOE executes; these process spaces are protected from interference from other processes except through the defined TOE interfaces. Processes are separate from each other, each with their own memory buffer and it is impossible for one process to directly access the memory of another. The OS and hardware support non-bypassability by ensuring that all access to resources designated to be encrypted is mediated through the TOE.

# 2 Conformance claims

## 2.1 Common Criteria conformance claim

The TOE is Common Criteria Version 3.1 Revision 4 (September 2012) Part 2 extended and Part 3 conformant at Evaluation Assurance Level 2 and augmented by ALC_FLR.2 – Flaw Reporting Procedures.

## 2.2 Protection Profile conformance claim

The TOE does not claim conformance to any Protection Profile.

# 3 Security problem definition

In order to clarify the nature of the security problem that the TOE is intended to solve, this section describes the following:

- Any known or assumed threats to the assets against which specific protection within the TOE or its environment is required.
- Any organisational security policy statements or rules with which the TOE must comply.
- Any assumptions about the security aspects of the environment and/or of the manner in which the TOE is intended to be used.

This chapter identifies assumptions as A.*assumption*, threats as T.*threat* and policies as P.*policy*.

## 3.1 Threats

The following are threats identified for the TOE and the information the TOE protects. The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides. The assumed level of expertise of the attacker for all the threats is unsophisticated.

The TOE addresses the following threats:

| THREAT | DESCRIPTION |
|---|---|
| T.KEY_MATERIAL_ COMPROMISE | An attacker can obtain unencrypted key material that the TOE has written to persistent memory, and use these values to gain access to user data. |
| T.PERSISTENT_INFORMATION | The Operational Environment can go into a power saving mode so that the data or keying material are left unencrypted in persistent memory. |
| T.KEYSPACE_EXHAUST | An unauthorized user may attempt a brute force attack to determine cryptographic keys or authorization factors to gain unauthorized access to data or TOE resources. |
| T.TSF_ COMPROMISE | A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted) to gain access to key material or user data. |
| T.UNAUTHORIZED_DATA_ACCESS | An unauthorized user that has access to the host computer may gain access to data for which they are not authorized according to the TOE security policy. |

**Table 9 – Threats Addressed by the TOE**

## 3.2 Organisational security policies

There are no organisational security policies applicable to the TOE.

## 3.3 Assumptions

This section describes the security aspects of the environment in which the TOE is intended to be used. The TOE is assured to provide effective security measures in a co-operative non-hostile environment only if it is installed, managed, and used correctly. The following specific conditions are assumed to exist in an environment where the TOE is employed. If the TOE is placed in an operational environment that does not meet these assumptions, the TOE may not be able to provide all of its security functionality. Assumptions can be on physical, personnel and connectivity of the operational environment.

| ASSUMPTION | DESCRIPTION |
|---|---|
| A.PASSWORD | An authorized administrator will be responsible for ensuring that passwords have sufficient strength and entropy to reflect the sensitivity of the data being protected. |
| A.PLATFORM_I&A | The TOE will be installed on a platform that supports individual user identification and authentication. This I&A functionality will remain unaffected by the TOE. |
| A.PROTECT_INTEGRITY | Authorized administrators will exercise due diligence in physically protecting the TOE, and will ensure that the IT environment will sufficiently protect against logical attacks. |
| A.SHUTDOWN | Authorized administrators will not leave the machine in a mode where sensitive information persists in non-volatile storage (e.g. power it down or enter a power managed state, such as a “hibernation mode”). |
| A.TRAINED_USERS | Authorized administrators are appropriately trained and follow all appropriate guidance documentation. |

**Table 10 – Assumptions**

# 4 Security objectives

The security objectives for the Target of Evaluation (TOE) and for the operational environment are derived from the threats and assumptions in Section 3.

## 4.1 Security objectives for the TOE

The IT security objectives for the TOE are addressed below. The TOE has to meet these objectives by satisfying the security functional requirements.

| OBJECTIVE | DESCRIPTION |
|---|---|
| O.ENCRYPT | The TOE must encrypt all data that are stored in specified files and removable media. |
| O.ACCESS | The TOE must permit access to protected information only to users who have been successfully authenticated, and who are permitted access in line with established policies. |
| O.PASSWORD | The TOE must ensure that passwords used to protect user data are of sufficient strength. |
| O.KEY_MATERIAL_COMPROMISE | The TOE must zeroize key material as soon as it is no longer needed to decrease the chance that such material could be used to expose protected information. |
| O.MANAGE | The TOE must provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use. |

**Table 11 – TOE Security Objectives**

## 4.2 Security objectives for the operational environment

The security objectives for the operational environment are addressed below:

| OBJECTIVE | DESCRIPTION |
|---|---|
| OE.PASSWORD_STRENGTH | An authorized administrator must ensure that passwords conform to appropriate guidance from the organisation using the TOE. |
| OE.PLATFORM_I&A | The Operational Environment must provide individual user identification and authentication mechanisms that operate independently of the TOE. |
| OE.TRAINED_USERS | Authorized administrators must be properly trained and follow all guidance for securing the TOE and authorization factors. |
| OE.PROTECT | The Operational Environment will protect itself and the TOE from external interference or tampering. |

**Table 12 – Operational Environment Security Objectives**

## 4.3 Security objectives rationale

This section provides the summary that all security objectives are traced back to aspects of the addressed assumptions and threats. The following table provides a high level mapping of coverage for each threat and assumption.

| | T.KEY_MATERIAL_ COMPROMISE | T.PERSISTENT_INFORMATION | T.KEYSPACE_EXHAUST | T.TSF_ COMPROMISE | T.UNAUTHORIZED_DATA_ACCESS | A.PASSWORD | A.PLATFORM_I&A | A.PROTECT_INTEGRITY | A.SHUTDOWN | A.TRAINED_USERS |
|---|---|---|---|---|---|---|---|---|---|---|
| O.ENCRYPT | | | X | | X | | | | | |
| O.ACCESS | | | | | X | | | | | |
| O.PASSWORD | | | X | | | | | | | |
| O.KEY_MATERIAL_COMPROMISE | X | X | | | | | | | | |
| O.MANAGE | | | | X | | | | | | |
| OE.PASSWORD_STRENGTH | | | X | | | X | | | | |
| OE.PLATFORM_I&A | | | | X | | | X | | | |
| OE.TRAINED_USERS | | X | | | | | | X | X | X |
| OE.PROTECT | | | | | | | | X | | |

**Table 13 – Mapping of Assumptions, Threats, and OSPs to Security Objectives**

The following table provides detailed evidence of the mapping between security objectives and assumptions.

| THREATS, POLICIES, AND ASSUMPTIONS | ADDRESSED BY | RATIONALE |
|---|---|---|
| T.KEY_MATERIAL_ COMPROMISE<br>An attacker can obtain unencrypted key material that the TOE has written to persistent memory, and use these values to gain access to user data. | O.KEY_MATERIAL_COMPROMISE<br>The TOE must zeroize key material as soon as it is no longer needed to decrease the chance that such material could be used to expose protected | The TOE minimizes the opportunity for an attacker to compromise plaintext key material while in memory by zeroizing such material as soon as it is no |

| THREATS, POLICIES, AND ASSUMPTIONS | ADDRESSED BY | RATIONALE |
|---|---|---|
| | information. | longer needed. |
| T.PERSISTENT_INFORMATION<br>The Operational Environment can go into a power saving mode so that the data or keying material are left unencrypted in persistent memory. | O.KEY_MATERIAL_COMPROMISE<br>The TOE must zeroize key material as soon as it is no longer needed to decrease the chance that such material could be used to expose protected information.<br>OE.TRAINED_USERS<br>Authorized administrators will be properly trained and follow all guidance for securing the TOE and authorization factors. | Users may leave the TOE unattended, or transport it, while in a power saving mode. The TOE must therefore minimize the risk that plaintext key material will be present when the TOE enters such a mode. Guidance must be given to help ensure that users close all sensitive files before the TOE enters a power saving mode. |
| T.KEYSPACE_EXHAUST<br>An unauthorized user may attempt a brute force attack to determine cryptographic keys or authorization factors to gain unauthorized access to data or TOE resources. | O.ENCRYPT<br>The TOE must encrypt all data that are stored in specified files and removable media.<br>O.PASSWORD<br>The TOE must ensure that passwords used to protect user data are of sufficient strength.<br>OE.PASSWORD_STRENGTH<br>An authorized administrator must ensure that passwords conform to guidance from the Enterprise using the TOE. | To avoid the possibility of a successful brute force attack the TOE must ensure that all sensitive data is encrypted. TOE administrators must enforce the use of good quality passwords in the TOE environment. Where the TOE has control over passwords being applied to protect user data, the TOE must ensure they are of sufficient strength, in line with established policy. |
| T.TSF_ COMPROMISE<br>A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted) to gain access to key material or user data. | O.MANAGE<br>The TOE must provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.<br>OE.PLATFORM_I&A<br>The Operational Environment must provide individual user identification and authentication mechanisms that operate independently of the TOE. | The operational environment provides identification and authentication that will help protect the TSF data and executables from unauthorized attack. |

| THREATS, POLICIES, AND ASSUMPTIONS | ADDRESSED BY | RATIONALE |
|---|---|---|
| T.UNAUTHORIZED_DATA_ACCESS<br>An unauthorized user that has access to the host computer may gain access to data for which they are not authorized according to the TOE security policy. | O.ENCRYPT<br>The TOE must encrypt all data that are stored in specified files and removable media.<br>O.ACCESS<br>The TOE must permit access to protected information only to users who have been successfully authenticated, and who are permitted access in line with established policies. | As determined by policy, all data must be encrypted, and access must be granted only to those who have been successfully authenticated. This provides protection against attackers with physical access to the host computer, or logical access but without the necessary credentials to access the data. |
| A.PASSWORD<br>An authorized administrator will be responsible for ensuring that passwords have sufficient strength and entropy to reflect the sensitivity of the data being protected. | OE.PASSWORD_STRENGTH<br>An authorized administrator must ensure that passwords conform to appropriate guidance from the organisation using the TOE. | Identification and authentication for access to data stored on the TOE host computer is carried out in the operational environment.<br>Administrators are relied on to ensure that passwords are of appropriate strength, and are in line with organizational guidance.<br>Note that in some cases passwords are under TOE control, and for those cases guidance is reinforced by TOE controls (see O.PASSWORD). |
| A.PLATFORM_I&A<br>The TOE will be installed on a platform that supports individual user identification and authentication. This I&A functionality will remain unaffected by the TOE. | OE.PLATFORM_I&A<br>The Operational Environment must provide individual user identification and authentication mechanisms that operate independently of the TOE. | The assumption calls for a platform that provides identification and authentication for TOE users. The objective is that this be provided. |
| A.PROTECT_INTEGRITY<br>Authorized administrators will exercise due diligence in physically protecting the TOE, and will ensure that the IT environment will sufficiently protect against logical attacks. | OE.TRAINED_USERS<br>Authorized administrators must be properly trained and follow all guidance for securing the TOE and authorization factors.<br>OR.PROTECT<br>The Operational Environment will protect itself and the TOE | The assumption calls for administrators to provide protection for the TOE against physical attacks, and for the IT environment to provide protection against logical attacks. Guidance must be provided |

| THREATS, POLICIES, AND ASSUMPTIONS | ADDRESSED BY | RATIONALE |
|---|---|---|
| | from external interference or tampering. | for this, and appropriate training must be given. |
| A.SHUTDOWN<br>An authorized administrator will not leave the machine in a mode where sensitive information persists in non-volatile storage (e.g. power it down or enter a power managed state, such as a "hibernation mode"). | OE.TRAINED_USERS<br>Authorized administrators must be properly trained and follow all guidance for securing the TOE and authorization factors. | It is assumed that users will not leave sensitive data files open and leave the TOE unattended when in a power managed state. Guidance must be provided and appropriate training given. |
| A.TRAINED_USERS<br>Authorized administrators are appropriately trained and follow all appropriate guidance documentation. | OE.TRAINED_USERS<br>Authorized administrators must be properly trained and follow all guidance for securing the TOE and authorization factors. | It is assumed that appropriate training will be provided for administrators. The objective is that this be provided in the operational environment. |

**Table 14 – Rationale for Mapping of Threats, Policies, and Assumptions to Objectives**

# 5 Extended Components Definition

## 5.1 Introduction

This section provides definitions for CC Part 2 extended components that are used within this ST.

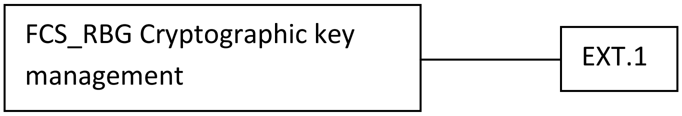
## 5.2 Extended components

### 5.2.1 Cryptographic operation: random bit generation (FCS_RBG_EXT)

**Family Behaviour**

This family is added to the class FCS. This family deals with generation of random bit streams in support of cryptographic operations

**Component Levelling**



FCS_RBG_EXT.1 requires generation of random bits in accordance with a selected standard.

**Management**: FCS_RGB_EXT.1

There are no management activities foreseen.

**Audit**: FCS_RGB_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: Failure of the activity.

**FCS_RGB_EXT.1 Random bit generation**

**Hierarchical to**: No other components.

**Dependencies**: None

FCS_RGB_EXT.1 .1 The TSF shall perform all random bit generation (RBG) services in accordance with [assignment: *method for random number generation*] seeded by an entropy source that accumulated entropy from [selection, one or both of: *a software-based noise source, a TSF-hardware-based noise source*].

FCS_RGB_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [selection, choose one of: *128 bits, 256 bits*] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

# 6 Security Requirements

The security requirements that are levied on the TOE and the IT environment are specified in this section of the ST.

## 6.1 Security functional requirements

The functional security requirements for this Security Target consist of the following components from Part 2 of the CC, and those that were explicitly stated, all of which are summarized in the following table:

| CLASS HEADING | COMPONENT | DESCRIPTION |
|---|---|---|
| Cryptographic support (FCS) | FCS_COP.1(1) | Cryptographic operation (AES) |
| | FCS_COP.1(2) | Cryptographic operation (RSA) |
| | FCS_COP.1(3) | Cryptographic operation (SHA) |
| | FCS_CKM.1(1) | Cryptographic key generation (spipe) |
| | FCS_CKM.1(2) | Cryptographic key generation (FRP) |
| | FCS_CKM.4 | Cryptographic key destruction |
| | FCS_RBG_EXT.1(1) | Extended: Random bit generation spipe |
| | FCS_RBG_EXT.1(2) | Extended: Random bit generation FRP |
| User data protection Class (FDP) | FDP_ACC.1 | Subset access control |
| | FDP_ACF.1 | Security attribute based access control |
| | FDP_RIP.1 | Subset residual information protection |
| Identification and authentication Class (FIA) | FIA_ATD.1 | User attribute definition |
| | FIA_UAU.2 | User authentication before any action |
| | FIA_SOS.1 | Verification of secrets |
| | FIA_UID.2 | User identification before any action |
| | FIA_USB.1 | User subject binding |
| Security management Class (FMT) | FMT_MOF.1 | Management of functions |
| | FMT_MTD.1(1) | Management of TSF data |

| CLASS HEADING | COMPONENT | DESCRIPTION |
|---|---|---|
| | FMT_MTD.1(2) | Management of TSF data |
| | FMT_SMF.1 | Specification of management functions |
| | FMT_SMR.1 | Security management roles |
| Protection of the TSF Class (FPT) | FPT_ITT.1 | Basic internal TSF data transfer protection |

**Table 15 – TOE Functional Components**

### 6.1.1 Cryptographic support (FCS)

#### *6.1.1.1 FCS_CKM.1(1) Cryptographic key generation (spipe)*

FCS_CKM.1.1(1) **Refinement:** The TSF shall generate cryptographic keys **for spipe communication using a Random Bit Generator as specified in FCS_RBG_EXT.1 and** specified cryptographic key sizes [*256 bits*] that meet the following: [*FIPS 186-2 (Change Note 1) for the ePO component and NIST SP 800-90 for the McAfee Agent component*].

#### *6.1.1.2 FCS_CKM.1(2) Cryptographic key generation (FRP)*

FCS_CKM.1.1(2) **Refinement:** The TSF shall generate **file and removable media** cryptographic keys **using a Random Bit Generator as specified in FCS_RBG_EXT.1 and** specified cryptographic key sizes [*256 bits*] that meet the following: [*no Standard*].

#### *6.1.1.3 FCS_CKM.4 Cryptographic key destruction*

FCS_CKM.4.1 The TSF shall destroy cryptographic keys **and critical security paramenters** in accordance with a specified cryptographic key destruction method [*zeroisation*] that meets the following: [*no standard*].

#### *6.1.1.4 FCS_COP.1(1) Cryptographic operation (AES)*

FCS_COP.1.1(1) The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES used in CBC and CFB mode*[3]] and cryptographic key sizes [*256 bits*] that meet the following: [*FIPS PUB 197, "Advanced Encryption Standard (AES)"*].

#### *6.1.1.5 FCS_COP.1(2) Cryptographic operation (RSA)*

FCS_COP.1.1(2) The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*RSA*] and cryptographic key sizes [*2048 bits*] that meet the following: [*PKCS#1 v2.1*].

[3] CBC = Cipher Block Chaining, CFB = Cipher Feedback

#### 6.1.1.6 FCS_COP.1(3) Cryptographic operation (SHA)

FCS_COP.1.1(3) The TSF shall perform [*cryptographic hashing*] in accordance with a specified cryptographic algorithm [*SHA-256*] and key sizes [*not applicable*] that meet the following: [*FIPS PUB 180-3, "Secure Hash Standard"* ].

#### 6.1.1.7 FCS_RGB_EXT.1 Random bit generation (1) spipe

FCS_RBG_EXT.1.1(1) The TSF shall perform all random bit generation (RBG) services in accordance with [*NIST Special Publication 800-90 using HMAC_DRBG*[4] *and FIPS Pub 186 (Change Notice 1)*[5]] seeded by an entropy source that accumulated entropy from [a software-based noise source].

FCS_RBG_EXT.1.2(1) The deterministic RBG shall be seeded with a minimum of [256 bits] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

#### 6.1.1.8 FCS_RGB_EXT.1 Random bit generation (2) FRP

FCS_RBG_EXT.1.1(2) The TSF shall perform all random bit generation (RBG) services in accordance with [*NIST Special Publication 800-90 using HMAC_DRBG*] seeded by an entropy source that accumulated entropy from [selection, one or both of: [a software-based noise source].

FCS_RBG_EXT.1.2(2) The deterministic RBG shall be seeded with a minimum of [256 bits] of entropy at least equal to the greatest bit length of the keys and authorization factors that it will generate.

### 6.1.2 User data protection (FDP)

#### 6.1.2.1 FDP_ACC.1 Subset access control

FDP_ACC.1.1 The TSF shall enforce the [*access control SFP*] on [*subjects: users; objects: files and removable media, processes, directories; operations: read, write, list*].

#### 6.1.2.2 FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the [*access control SFP*] to objects based on the following:
[*subjects: users;*
*objects: data (controlled files and removable media), processes directories;*
*security attributes: pathname, file extension, process name*].

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

[4] MA
[5] ePO

a) *A user shall be able to decrypt and read data only if that access is consistent with the assigned policy;*
b) *A user shall be able to encrypt data only if that is consistent with the assigned policy;*
c) *A user shall be able to list encrypted directories only if that is consistent with the assigned policy;*
d) *A user shall have access to files created by an application (process name or file extension) only if that is consistent with the assigned policy;*
e) *A user shall be able to write encrypted data to optical media or to ISO images that can be authenticated without the need to install the FRP client only if that is consistent with the assigned policy*].

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

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

#### *6.1.2.3 FDP_RIP.1 Subset residual information protection*

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

### 6.1.3 Identification and authentication (FIA)

#### *6.1.3.1 FIA_ATD.1 User attribute definition*

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

*a. ePO User name;*

*b. Enabled or disabled;*

*c. Authentication configuration (either local ePO authentication credentials or Windows authentication); and*

d. *Permission Sets*].

*Application Note: The TOE maintains security attributes for ePO users. Windows maintains security attributes for Workstation Users.*

#### *6.1.3.2 FIA_SOS.1 Verification of secrets*

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet

[*a. Total password length (4-40 characters)*
*b. Lowercase characters (0-15 characters)*
*c. Uppercase characters (0-15 characters)*
*d. Total alphabetic characters (0-15 characters)*

*e. Numeric characters (0-15 characters)*

*f. Special characters (0-15 characters)*].

#### *6.1.3.3 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.

#### *6.1.3.4 FIA_UID.2 User identification before any action*

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

#### *6.1.3.5 FIA_USB.1 User-subject binding*

FIA_USB.1.1 The TSF shall associate the following **ePO** user security attributes with subjects acting on behalf of that user: [

*a. ePO User name;*

*b. Permissions*].

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of **ePO** users: [*user security attributes are bound upon successful login with a valid ePO User Name*].

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of **ePO** users: [*user security attributes do not change during a session*].

*Application Note: The TOE binds security attributes to subjects for ePO sessions. Windows binds security attributes to subjects for workstation sessions. Permissions are determined by the union of all permissions in any permission set associated with a user. If the security attributes for a user are changed while that user has an active session, the new security attributes are not bound to a session until the next login.*

### 6.1.4 Security management (FMT)

#### *6.1.4.1 FMT_MOF.1 Management of security functions behaviour*

FMT_MOF.1.1 The TSF shall restrict the ability to [determine the behaviour of, disable, enable] the functions [

*a. Operation of an instance of FRP on a workstation]*

to [*an authorized administrator*].

### 6.1.4.2 FMT_MTD.1(1) Management of TSF data

FMT_MTD.1.1(1) The TSF shall restrict the ability to [create, read, modify, delete] the [

*a) Enforcement of FRP policies on a system*

b) *User interface option settings*]

to [*an authorized administrator*].

### 6.1.4.3 FMT_MTD.1(2) Management of TSF data

*Application Note: The TSF data referenced in this SFR corresponds to the policies identified in* Table 8 – TOE Data*.*

FMT_MTD.1.1(2) The TSF shall restrict the ability to [query, modify, delete, *create and use*] the [*TSF data identified in the following table*] to [*a user with the permissions identified in the following table or an ePO Administrator*].

| TSF Data | Associated Permission | Operations Permitted |
|---|---|---|
| | | |
| | | |
| | | |
| ePO User Accounts | n/a (only allowed by an ePO Administrator) | Query, create, delete and modify |
| FRP Key Server | View key server – permits users only to view FRP keys | View |
| | Manage Key Server – permits users to create and manage FRP keys | View and modify |
| FRP Policy Permissions | View policy and task settings | View |
| | View and change policy and task settings | View and modify |
| Permission Set | n/a (only allowed by an ePO administrator) | Query, create, duplicate, delete and modify |
| | | |
| | | |
| | | |
| Registered Servers – LDAP | View registered servers | View |
| Systems | View "System Tree" tab | Query |
| | Actions | Wake up Agents; view Agent Activity Log<br>Edit System Tree groups and systems<br>Deploy agents |

| TSF Data | Associated Permission | Operations Permitted |
|---|---|---|
| System tree access | Access nodes and portions of the System Tree | Access nodes and portions of the System Tree |

**Table 16 - TSF Data Access Permissions**

#### 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: [

*a) Configure operation of the TOE on workstations,*

*b) ePO user account management,*

*c) Permission set management,*

*d) Policy management,*

*e) System tree management,*

*f) Query and Report management*].

#### 6.1.4.5 FMT_SMR.1 Security roles

FMT_SMR.1.1 The TSF shall maintain the roles [administrator *and ePO user assigned to any of the following permissions or combinations of permission sets:*

*a.* Audit Log
*b.* Client Events
*c.* Dashboards
*d.* FRP Key Server
*e.* FRP Policy Permissions
*f.* Queries and Reports
*g.* Registered Servers
*h.* Systems
*i.* System Tree access].

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

### 6.1.5 Protection of the TSF (FPT)

#### 6.1.5.1 FPT_ITT.1 Internal TOE TSF data transfer

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

## 6.2 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 2 (EAL2) augmented by ALC_FLR.2. The assurance components are summarized in the following table:

| CLASS HEADING | CLASS_FAMILY | DESCRIPTION |
|---|---|---|
| ADV: Development | ADV_ARC.1 | Security architecture description |
| | ADV_FSP.2 | Security-enforcing functional specification |
| | ADV_TDS.1 | Basic design |
| AGD: Guidance Documents | AGD_OPE.1 | Operational user guidance |
| | AGD_PRE.1 | Preparative procedures |
| ALC: Lifecycle Support | ALC_CMC.2 | Use of a CM system |
| | ALC_CMS.2 | Parts of the TOE CM coverage |
| | ALC_DEL.1 | Delivery procedures |
| | ALC_FLR.2 | Flaw reporting procedures |
| ATE: Tests | ATE_COV.1 | Evidence of coverage |
| | ATE_FUN.1 | Functional testing |
| | ATE_IND.2 | Independent testing - sample |
| AVA: Vulnerability Assessment | AVA_VAN.2 | Vulnerability analysis |

**Table 17 – Security Assurance Requirements**

## 6.3 CC component hierarchies and dependencies

This section of the ST demonstrates that the identified SFRs include the appropriate hierarchy and dependencies. The following table lists the TOE SFRs and the SFRs each are hierarchical to, dependent upon and any necessary rationale.

| SFR | HIERARCHICAL TO | DEPENDENCY | RATIONALE |
|---|---|---|---|
| FCS_CKM.1(1) | None | FCS_CKM.2 or FCS_COP.1, FCS_CKM.4 | Met by inclusion of FCS_COP.1(1) and FCS_CKM.4 |
| FCS_CKM.1(2) | None | FCS_CKM.2 or FCS_COP.1, FCS_CKM.4 | Met by inclusion of FCS_COP.1(2) and FCS_CKM.4 |
| FCS_CKM.4 | None | FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1 | Met by inclusion of FCS_CKM.1(1) &(2) |
| FCS_COP.1(1) | None | FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1, FCS_CKM.4 | Satisfied by inclusion of FCS_CKM.1(1)and FCS_CKM.4 |

| SFR | HIERARCHICAL TO | DEPENDENCY | RATIONALE |
|---|---|---|---|
| FCS_COP.1(2) | None | FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1, FCS_CKM.4 | Satisfied by inclusion of FCS_CKM.1(2) and FCS_CKM.4 |
| FCS_COP.1(3) | None | FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1, FCS_CKM.4 | Satisfied by inclusion of FCS_CKM.1(1) and FCS_CKM.4 |
| FCS_RGB_EXT.1(1) | None | None | N/A |
| FCS_RGB_EXT.1(2) | None | None | N/A |
| FDP_ACC.1 | None | FDP_ACF.1 | Satisfied |
| FDP_ACF.1 | None | FDP_ACC.1,<br>FMT_MSA.3 | Satisfied by inclusion of FDP_ACC.1.<br>FMT_MSA.3 is not included as the setting of secure default values for controlled data is not required. |
| FDP_RIP.1 | None | None | N/A |
| FIA_ATD.1 | None | None | N/A |
| FIA_SOS.1 | None | None | N/A |
| FIA_UAU.2 | FIA_UAU.1 | FIA_UID.1 | Satisfied by inclusion of FIA_UID.2 |
| FIA_UID.2 | FIA_UID.1 | None | N/A |
| FIA_USB.1 | None | FIA_ATD.1 | Satisfied |
| FMT_MOF.1 | None | FMT_SMF.1,<br>FMT_SMR.1 | Satisfied<br>Satisfied |
| FMT_MTD.1(1) | None | FMT_SMF.1,<br>FMT_SMR.1 | Satisfied<br>Satisfied |
| FMT_MTD.1(2) | None | FMT_SMF.1,<br>FMT_SMR.1 | Satisfied<br>Satisfied |
| FMT_SMF.1 | None | None | N/A |
| FMT_SMR.1 | None | FIA_UID.1 | Satisfied by inclusion of FIA_UID.2 |
| FPT_ITT.1 | None | None | N/A |

**Table 18 – TOE SFR Dependency Rationale**

## 6.4 Security requirements rationale

This section provides rationale for the Security Functional Requirements demonstrating that the SFRs are suitable to address the security objectives

### 6.4.1 Security functional requirements for the TOE

The following table provides a high level mapping of coverage for each security objective:

| | O.ENCRYPT | O.ACCESS | O.PASSWORD | O.KEY_MATERIAL_COMPROMISE | O.MANAGE |
|---|---|---|---|---|---|
| FCS_COP.1(1) | X | | | | |
| FCS_COP.1(2) | X | | | | |
| FCS_COP.1(3) | X | | | | |
| FCS_CKM.1(1) | X | | | | |
| FCS_CKM.1(2) | X | | | | |
| FCS_CKM.4 | X | | | | |
| FCS_RBG_EXT.1(1) | X | | | | |
| FCS_RBG_EXT.1(2) | X | | | | |
| FDP_ACC.1 | | X | | | |
| FDP_ACF.1 | | X | | | |
| FDP_RIP.1 | | | | X | |
| FIA_ATD.1 | | X | | | X |
| FIA_UAU.2 | | X | | | X |
| FIA_UID.2 | | X | | | X |
| FIA_SOS.1 | | X | X | | |
| FIA_USB.1 | | X | | | X |
| FMT_MOF.1 | | X | | | |
| FMT_MTD.1(1) | | X | | | X |
| FMT_MTD.1(2) | | | | | X |
| FMT_SMF.1 | | X | | | X |
| FMT_SMR.1 | | X | | | X |

| | O.ENCRYPT | O.ACCESS | O.PASSWORD | O.KEY_MATERIAL_COMPROMISE | O.MANAGE |
|---|---|---|---|---|---|
| FPT_ITT.1 | | X | | | X |

Table 19 – Mapping of TOE SFRs to Security Objectives

The following table provides detailed evidence of coverage for each security objective:

| OBJECTIVE | REQUIREMENTS THAT ADDRESS THE OBJECTIVE | SFR AND RATIONALE |
|---|---|---|
| O.ENCRYPT | FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.4, FCS_RGB_EXT.1(1), FCS_RGB_EXT.1(2) | The TOE uses AES to encrypt specified data (FCS_COP.1(1)). Keys are generated and deleted as required (FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.4, FCS_RGB_EXT.1(1), FCS_RGB_EXT.1(2)). Keys generated on ePO are transferred to FRP clients in encrypted form (FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3)). |
| O.ACCESS | FDP_ACC.1, FDP_ACF.1, FIA_ATD.1, FIA_UAU.2, FIA_UID.2, FIA_SOS.1, FIA_USB.1, FMT_MOF.1, FMT_MTD.1(1), FMT_MTD.1(2), FMT_SMR.1, FPT_ITT.1 | This objective covers both access to management functions on ePO, and to the user data stored on the FRP clients. Access to management functions (specified in FMT_MOF.1, FMT_MTD.1(1), FMT_MTD.1(2)) is restricted to authorized users (FIA_ATD.1, FIA_UAU.2, FIA_UID.2, FIA_SOS.1, FIA_USB.1). Access to user data requires a successful user logon (outside the scope of the TOE) and enforcement of the defined policies (FDP_ACC.1 and FDP_ACF.1). Keys are required to be protected in transit by FPT_ITT.1. |
| O.PASSWORD | FIA_SOS.1 | FIA_SOS.1 requires that rules can be set for password quality. Guidance specifies a minimum standard to be enforced using this mechanism. |

| OBJECTIVE | REQUIREMENTS THAT ADDRESS THE OBJECTIVE | SFR AND RATIONALE |
|---|---|---|
| O.KEY_MATERIAL _COMPROMSE | FDP_RIP.1 | The objective to provide protection against compromise of keys is met through the requirement to ensure residual information is removed from volatile memory when no longer required. |
| O.MANAGE | FIA_ATD.1, FIA_UAU.2, FIA_UID.2, FIA_USB.1, FMT_MTD.1(1), FMT_MTD.1(2), FMT_SMF.1, FMT_SMR.1, FPT_ITT.1 | Access to management functions on ePO is restricted to authorized users (FIA_UAU.2, FIA_UID.2, FIA_ATD.1, FIA_USB.1). Access is controlled using roles (FMT_SMR.1). Management functions are specified in FMT_MTD.1(1), FMT_MTD.1(2) and FMT_SMF.1. The TOE provides protection for the transfer of key material and policies from ePO to FRP (FPT_ITT.1). |

**Table 20 – Rationale for Mapping of TOE SFRs to Objectives**

### 6.4.2 Rationale for TOE Security assurance requirements

The general level of assurance for the TOE (EAL2 plus ALC_FLR.2) is consistent with current best commercial practice for IT development and provides a product that is competitive against non-evaluated products with respect to functionality, performance, cost, and time-to-market. The TOE assurance also meets current constraints on widespread acceptance, by expressing its claims against EAL2 augmented by ALC_FLR.2 from part 3 of the Common Criteria.

# 7 TOE Summary Specification

## 7.1 Cryptographic support

FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_RBG_EXT.1(1), FCS_RBG_EXT.1(2)

Data is encrypted using AES 256-bit, operating in CBC and CFB modes.

All regular and user personal keys are generated by ePO and exported to instances of FRP. FRP keys are created on the ePO server within the FRP extension. FRP uses the default crypto provider that is shipped with ePO. In FIPS mode ePO uses crypto-J from RSA.

As soon as they are created, the keys are protected using McAfee Foundation Service (MFS) API, part of the low level infrastructure that underpins ePO, and then written to the data base. The buffers are not zeroised in the server, but are rewritten immediately with their encrypted version. Each key has a Globally Unique Identifier (GUID) that can identify it. Administrators assign key GUIDs to Grant Keys policies, and these keys are then assigned to systems. On receiving the key GUIDs, FRP requests for the keys via the data channel. All requests and responses in the data channel are further encrypted by a hard coded AES-256 key. The same hard coded key is present both on the extension as well as the client. This mechanism is more to obfuscate the key rather than encrypt it along the channel. The data channel is already protected using TLS v1.0 by McAfee Agent using the McAfee proprietary s-pipe protocol.

Once FRP receives the key, encrypted with the hard-coded obfuscation key, FRP stores it in the per user KeyCache (which is a file on the file system) and then loads the keys to the vault driver. Soon after the key is loaded to the vault driver, the key secret is removed from core process and core service memory using RTLSecureZeroMemory and SecureZeroMemory APIs. All this is done while the keys are always in-memory of the core service and core process.

The key cache itself is protected using Microsoft DPAPI, CryptProtectData and CryptUnprotectData, which function in FIPS mode when the OS itself is running in FIPS mode. All keys are protected by the user password.

The FRP client is required to generate user local keys. It is also required to perform encryption and decryption operations on user data. The FRP client makes use of the McAfee Core Cryptographic Module (MCCM) User (Cert.#2239) and Kernel (Cert.#2223) FIPS 140‐2 cryptographic modules.

In FIPS mode, ePO uses OpenSSL  v0.9.8.6 with FIPS module v1.2.3 (FIPS 140-2 certificate #1051) for TLS 1.1. McAfee Agent uses RSA BSAFE Crypto-C Micro Edition v4.0.1 (FIPS 140-2 certificate #2097) to provide cryptographic services for this link. McAfee affirms that the cryptographic modules have been implemented in accordance with their FIPS 140 security policies, and when the TOE is configured in FIPS mode the cryptographic functions operate as intended.

Random number generation for MA and FRP is done in accordance with NIST SP 800-90 using HMAC_DRBG, seeded using entropy from a software based noise source. For ePO, random number generation to support the spipe communication link to MA uses FIPS 186-2 (Change Note 1).

## 7.2 User data protection

FDP_ACC.1, FDP_ACF.1, FDP_RIP.1

On Windows platforms, FRP supports three approaches to encryption:

1. Explicit encryption;
2. Encryption via desktop integration modules;
3. On the fly, transparent encryption via FRP driver for file and folder encryption policies.

In FRP folders are not encrypted, but just marked for encryption to let the FRP modules know that the contents (files) within the folder need to be encrypted. This is done by creating a ".cekey" file which will contain the key ID to be used for encrypting the folder. The .cekey file is a hidden system file.

The files are encrypted by the FRP driver via the FRP vault driver. This (vault driver) is the only module that performs encryption (via the MCCM cryptomodule) in the kernel mode. When a file gets encrypted, FRP adds a 512 byte header which also contains the GUID of the key with which it was encrypted.

FRP has a file system filter driver. All Input/Output Request packets (IRPs) definitely pass through the FRP driver irrespective of the file being opened/saved being plain text or cipher text. The driver evaluates policies configured for the user/system as well as the existing file status, and determines whether or not to perform on the fly encryption/decryption. All API/applications performing IO will inevitably go through the driver, and hence the user receives a transparent encryption experience. To determine if a file is encrypted or not, FRP appends a special character to the end of file name. This and the key GUID stored in the first 512 bytes of the key allows FRP driver to provide this service.

On user login, the FRP core process fetches the keys from the user specific key cache. The key cache stores all of the keys assigned to the system and user. This is then decrypted using Microsoft DP API that uses the logged in user credentials to unlock the cache. The keys are then loaded into the vault driver and immediately removed (zeroised) from the core process memory.

On power down, the vault driver simply unloads all keys. No encryption or storage is required as the keys are encrypted and stored in the key cache when FRP obtains them from ePO. There is no encryption done at this point.

Only the FRP core service, core process, vault driver and MCCM cryptomodule access the keys in plain text format. Both the core service and process, and the MCCM, zeroise the key secret immediately after loading it into the vault driver. The vault driver always holds keys in-memory so they are removed on system shut down. All other modules of FRP, including the data channel, always handle either encrypted keys or only the key GUID.

The key is never written to disk in plaintext. The only place the key persists is in the key cache (on file), which is always protected by the Microsoft DPAPI documented above.

In FIPS mode ePO uses crypto-J from RSA. As soon as a key is created, it is protected using MFS API and then written to the data base. The buffers are not zeroised in the server, but are rewritten immediately with their encrypted version. Each key has a unique GUID that can identify it. Administrators assign key GUID to Grant Keys policies, and these keys are then assigned to systems. On receiving the key GUIDS, the FRP client requests the keys via the data channel. All requests and responses in the data channel are further encrypted by a hard coded key using AES-256 encryption. The same hard coded key is present both on the ePO extension and on the client. This mechanism is intended to obfuscate the key rather than encrypt it along the channel. The data channel is already protected by MA using the McAfee proprietary s-pipe protocol (TLSv1.0).

Once FRP receives the key, encrypted with the hard-coded obfuscation key, FRP stores it in the per user Key Cache (which is a file on the file system), and then loads the keys to the vault driver. Soon after the key is loaded to the vault driver, the key secret is removed from core process and core service memory using the SecureZeroMemory API.

The key cache itself is protected using Microsoft DPAPI, CryptProtectData and CryptUnprotectData which function in FIPS mode if the OS itself is running in FIPS mode.

## 7.3 Identification and authentication

FIA_ATD.1, FIA_UAU.2, FIA_UID.2, FIA_USB.1, FIA_SOS.1

### 7.3.1 ePO Administrator Authentication

Identification of authorized administrators is enforced by the TOE. Authorized administrators must log in to the ePO administrative interface with a valid user name and password via a GUI before any access is granted by the TOE to TOE functions or data. When the credentials are presented by the user, ePO determines if the user name is defined. If not, the login process is terminated and the login GUI is redisplayed with a generic authentication error message.

The TOE determines the authentication method configured for the administrator with the user name entered and handles the password accordingly. If Windows authentication is configured, the username and a hash of the password entered by the user are passed to the Windows operating system for verification; therefore, the TOE IT Environment handles the authentication of users. If local ePO password authentication is configured then the TOE is responsible for performing the verification of the password entered. If the password is successfully validated for the user name, the TOE grants access to authorized TOE functionality. If the password is not validated, the login GUI is redisplayed to the user with a generic authentication error message.

For each defined user account, the following information is configured:

- User name
- Whether authentication for this user is to be performed by ePO or Windows
- Permission set(s) granted to the user

Upon successful login and each consecutive action taken that causes a GUI refresh, the permission sets are bound. Those attributes remain fixed until an action causes the GUI to refresh. If the attributes for a logged-in user are changed, those changes will not be bound to a subject until the next GUI action by that user.

### 7.3.2 FRP Client User Authentication

The FRP client depends on Windows authentication to determine the logged-in (non-privileged) user at the endpoint. This identification is neither generated nor configurable from the FRP product.

There are FRP modules such as User Local Keys, Self-Extractor and features related to removable media that are password based. For these modules FRP provides policy to setup password rules. The parameters below can be configured with a maximum password length of 40 characters.

- Total password length (4-40)
- Lowercase characters (0-15)
- Uppercase characters (0-15)
- Total alphabetic characters (0-15)
- Numeric characters (0-15)
- Special characters (0-15)

The salt is generated using MCCM HMAC DRBG, and then a KEK is created using the PBKDF algorithm.

The validity of the authorization factor is checked on input in a manner that does not expose any key material. At the time of container creation:

- Convert password to key using salt and PBKDF
- Create a new random array, this is 32 bytes of random data
- Calculate CRC32 checksum and store in the end of the vector
- Encrypt the vector and store it in the vector itself

At the time of verification:

- Convert user given password to key using salt and PBKDF
- Decipher the Vector
- Verify checksum to ensure password correctness

A user can change the authorization factor:

- The password is validated as above
- The container key stored in the header is decrypted using old password
- The new password is converted to key using salt and PBKDF
- The decrypted container key is encrypted using the new key and saved to container header

Data recovery for removable media containers and the user local keystore is possible if the authorization factor is lost. At the time of initialization, the passphrase is converted to a key using PBKDF. The actual key that is used to encrypt the MFRP container is then generated. This key is encrypted once with each recovery method and saved in the header. On recovery, the recovery information is used to unlock the actual container key. The recovery methods available are by password, recovery key and certificate. Recovery for user local keys and removable media containers can be policy controlled. Other modules cannot be recovered. The policies are configurable only by an administrator. User local keys are not configurable even by administrators. Setting up a recovery key is mandatory.

## 7.4 Management (MGMT)

FMT_SMF.1, FMT_SMR.1, FMT_MOF.1, FMT_MTD.1

The TOE's Management Security Function provides support functionality that enables an authorized administrator to configure and manage TOE components. Management of the TOE may be performed via the ePO GUI.

ePO management permissions are defined per-user. Configuring the Administrator permission set to an ePO account implicitly grants all user permissions to that user (no other permission sets can be applied to a user assigned to the administrator permission set). Upon successful authentication (see section 7.3.1 above), the union of all the permissions from the permission sets from the user account configuration are bound to the session, along with the user name. Those attributes remain fixed for the duration of the session (until the user logs off).

The TOE provides functionality to manage the following:

1. Enable, disable and configure operation of the TOE on workstations ,
2. ePO User Account management,
3. Permission Set management,
4. Policy Management,

5. System tree management,
6. Report and Query management.

Each of these items is described in more detail in the following sections.

### 7.4.1 FRP Operation

FMT_SMF.1, FMT_MOF.1, FMT_MTD.1(1)

Policies can be deployed from ePO to an instance of FRP, and enabled and disabled remotely. The policies deployed to an FRP client enforce a particular FRP configuration at the managed system.  The configuration options for the operation of the FRP Client include:

- Allow explicit encryption
- Allow explicit decryption
- Enable padlock visibility
- Enable search encrypted objects
- Allow creation of self-extractor
- Enable sending of encrypted email attachments
- User Local key options
- File encryption options (encrypt all files of a specified file extension)
- Folder encryption options (encrypt all files in specified folder)
- Granting of keys for use on FRP client
- Enable network encryption
- Password rules
- Removable media protection levels (allow/enforce offsite access or onsite access only)
- CD/DVD protection levels (allow/enforce offsite access or onsite access only)

### 7.4.2 ePO user account management

FMT_SMF.1, FMT_MTD.1(2)

Each user authorized for login to ePO must be defined with ePO.  Only ePO Administrators may perform ePO user account management functions (create, view, modify and delete). For each defined account, the following information is configured:

1. User name
2. Enabled or disabled
3. Whether authentication for this user is to be performed by ePO or Windows
4. Permission sets granted to the user

One or more permission sets may be associated with an account.  ePO users assigned to the “Administrator “permission set are granted all permissions, and referred to as ePO administrators..

Permissions exclusive to those users assigned to the administrator Permission Set (which cannot be modified) include:

1. Change server settings.
2. Create and delete user accounts and groups.
3. Create, delete, and assign permission sets.

### 7.4.3 Permission set management

FMT_SMF.1, FMT_MTD.1(2)

A permission set is a group of permissions that can be granted to any users by assigning it to those users' accounts. One or more permission sets can be assigned to any users who is not already assigned to the Administrator permission set.

Permission sets only grant rights and access — no permission set ever removes rights or access. When multiple permission sets are applied to a user account, they aggregate. For example, if one permission set does not provide any permissions to server tasks, but another permission set applied to the same account grants all permissions to server tasks, that account has all permissions to server tasks.

Administrators may create, view, modify and delete permission sets. Each permission set has a unique name so that it can be appropriately associated with ePO users.

When a permission set is created or modified, the permissions granted via the permission set may be configured by an Administrator.

### 7.4.4 Policy management

FMT_SMF.1, FMT_MTD.1(1)

FRP policies are configured on ePO, and automatically distributed to the client systems running FRP. The policies determine what encryption/decryption operations are performed on the systems. Permissions relevant to FRP policies are:

1. View FRP policy permissions
2. View and change FRP policy permissions
3. View FRP key server permissions
4. View and change FRP key server permissions

Policies covering the following areas related to FRP may be configured (full details may be found in the Product Guide):

- Options for explicit encryption/decryption
- Creation of self-extractors
- Encrypted email attachments

- Folder paths for encryption
- File extensions/process names for encryption
- Encryption of USB media, floppy disks, CD/DVDs
- Encryption options
- Use of keys
- Encryption of network locations
- User local keys
- Password rules

### 7.4.5 System tree management

FMT_SMF.1, FMT_MTD.1(2)

The system tree organizes managed systems in units for monitoring, assigning policies, scheduling tasks, and taking actions. The system tree is a hierarchical structure that allows systems to be organized within units called groups.

Groups have these characteristics:

1. Groups can be created by Administrators.
2. A group can include both systems and other groups.
3. Groups are modified or deleted by an Administrator.

The system tree root includes a Lost&Found group. Depending on the methods for creating and maintaining the system tree, the server uses different characteristics to determine where to place systems. The Lost&Found group stores systems whose locations could not be determined.

The Lost&Found group has the following characteristics:

1. It can't be deleted;
2. It can't be renamed;
3. Its sorting criteria can't be changed (although sorting criteria can be provided for the subgroups created within it);
4. It always appears last in the list and is not alphabetized among its peers;
5. All users with view permissions to the system tree can see systems in Lost&Found;
6. When a system is sorted into Lost&Found, it is placed in a subgroup named for the system's domain, and if no such group exists, one is created.

Child groups in the system tree hierarchy inherit policies set at their parent groups. Inheritance is enabled by default for all groups and individual systems that you add to the system tree. Inheritance may be disabled for individual groups or systems by an Administrator.  Inheritance can be broken by applying a new policy at any location of the system tree (provided a user has appropriate permissions). Users can lock policy assignments to preserve inheritance.

User permissions in the Systems category that are relevant to this information are:

1. View the “system tree” tab;
2. Edit system tree groups and systems.

Systems may be deleted or moved between groups by an Administrator or users with both the “View the “system tree” tab” and “Edit system tree groups and systems” permissions.  User access to groups in the system tree is controlled by individual check boxes in the permission sets for the system tree.

### 7.4.6 Query and Report management

FMT_SMF.1, FMT_MTD.1(2)

Users may create, view, modify, use and delete queries/reports based upon their permissions. Permissions associated with queries and reports are:

1. Use public groups — Grants only permission to run queries or reports in public groups. Users granted this level of permission cannot create or edit their own queries.
2. Use public groups; create and edit private queries/reports — Grants permission to run any queries and reports in public groups, as well as the ability to create and edit queries with the Query builder wizard. This option not allow a user to add any queries or reports to public groups, or to edit any queries or reports in public groups..
3. Edit public groups; create and edit private queries/reports; make private queries/reports public — Grants full permission to the queries and Reports page, allowing the user to run and edit any queries and reports in public groups, create and modify queries with the Query Builder wizards.

## 7.5 Protection of the TSF

FPT_ITT.1

The TOE provides protection for the transfer of key material and policies from ePO to FRP.

Policies are transferred via the MA policy channel. This channel is protected by MA using s-pipe (MA security for communication). Once the client receives the policies, the client requests ePO for the keys using the MA data channel. This request contains a key and a context (user/system for which the request is being made). The FRP extension inside ePO receives the requests and validates that the key request is genuine. This validation is done by checking the grant key policy assignments to ascertain that the key was indeed assigned to the user/system and hence the request was received.

Once the FRP extension validates the key request as genuine, it sends down the key to the FRP client. The FRP extension obfuscates the key before sending it down the data channel, which is then unobfuscated. The data channel itself is protected by the MA s-pipe security similar to the policy and events channel.