Protection Profile for Software Full Disk Encryption

Mitigating the Risk of a Lost or Stolen Hard Disk

31 March 2014

Version 1.1

Table of Contents

Revision History

Version	Date	Description
1.0	14 February 2013	Initial release
1.1	31 March 2014	Changes related to first evaluation using this PP

1 Introduction to the PP

1.1 PP Overview of the TOE

This PP addresses the threat that an adversary will obtain a lost or stolen hard disk (e.g., a disk contained in a laptop or a portable external hard disk drive) containing sensitive data. The Target of Evaluation (TOE) defined in this Protection Profile (PP) is for a software full disk encryption product that encrypts the data on the hard disk device. As defined by NIST: "*Full Disk Encryption (FDE)*, also known as whole disk encryption, is the process of encrypting all the data on the hard drive used to boot a computer, including the computer's OS, and permitting access to the data only after successful authentication to the FDE product."[1] Note that software encryption products will leave a portion of the drive unencrypted for the Master Boot Record (MBR) and the initial bootable partition. For this Protection Profile, the term "disk encryption" will be interpreted as per the NIST definition of full disk encryption modified to allow software disk encryption products to leave a portion of the drive unencrypted for the MBR and bootable partition so long as no information is written there that could contain user data.

1.1.1 Usage and major security features of TOE

The TOE is used to protect data at rest. The set of objectives and security functional requirements is limited to a device (generally a laptop) that has been lost or stolen while powered off without any prior access by an adversary.

The hard disk is encrypted using a data encryption key (DEK). The DEK is masked using a key encryption key (KEK). The KEK can be derived from multiple components (referred to as submasks, which are derived from authorization factors) or obtained from a single submask. The foremost security objective of encrypting the storage devices is to force an adversary to perform a cryptographic exhaust against a prohibitively large key space.

TOEs that conform to this PP implement primary functions. The authorization function establishes and then collects authorization factors for users of the TOE to form the KEK. The disk encryption function is responsible for encrypting and decrypting all data written to the storage device using the DEK. The sponsor of the evaluation is responsible for the delivery of all of the information required by evaluation team to perform the assurance activities.

The vendor is additionally required to provide configuration guidance (AGD_PRE, AGD_OPR) to correctly install and administer the TOE for every Operational environment supported (for example, for every O/S supported by the product).

1.1.2 Authorization and Authentication

An authorized user of the hard disk provides one or more authorization factors when the computer is booted. These authorization factors determine whether the user is authorized to access the data on the

[1] NIST, "GUIDE TO STORAGE ENCRYPTION TECHNOLOGIES FOR END USER DEVICES", NIST Special Pub 800-111, November 2007.

disk drive. Authorization factors are not required to be unique to individual users. In other words, authorization factors for the disk encryption function are only being required to establish that the possessor is in the community of users authorized to access information stored on the hard disk.

After a user provides the correct authorization factors, the operating system will be decrypted and in most cases will present to the user the usual operating system login prompt. Identification and authentication of the user to the underlying OS, as well as with respect to the TOE management functions, is discussed in subsequent sections.

An authorization factor must consist of one of the following:

- An administrator-provided passphrase; or
- A bit string contained on an token external to the platform (e.g., a USB device)—defined as an external token authorization factor; or
- A combination of a passphrase and external token authorization factor;

And in addition to one of the above it may consist of:

- A Security Target (ST) author defined authorization factor (e.g. a bit string protected in a TPM and protected by a pin (which has anti-hammer protections incorporated)—defined as a TPM-protected authorization factor).

If the ST author defines additional authorization factors, they must be fully documented and cannot diminish the strength of the passphrase and/or external token authorization factors. All authorization factors must be conditioned such that they provide submasks that are the same size (bit length) as the key they are protecting, and must be combined using an XOR function to produce the KEK.

Passphrase authorization factors should be passphrases generated by an administrator using guidance contained in the TOE documentation as well as guidance provided by the user's enterprise. The TOE must support passphrases of at least 64 characters; this is to accommodate passphrases composed of several words of varying lengths (e.g., "This i$ a pr3tty s3cure PassPhraze"). Once the passphrase is entered by the user, the TOE conditions it using either a hash function, or an approved key derivation function that meets NIST 800-132 to produce the submask to be used as an input for the KEK.

The TPM authorization factor and the external token authorization factor are both just bit strings provided by the TOE; neither the external token itself nor the TPM are considered part of the TOE. They are distinguished in this PP, however, because there are additional operational considerations for the external token since it is not part of the computing platform (it is of course allowable to use an external token that contains a TPM with a pin-protected authorization factor, and in this case all of the assumptions for both authorization-factor providers would have to be met). In this PP, where an assumption or comment applies to both the TPM authorization factor and the external token authorization factor, the term "external authorization factor" is used.

The external authorization factors do not have to be generated by the TOE. If they are, the appropriate requirements from Appendix C must be included to specify that the authorization factor is generated by the TOE using a FIPS-approved random bit generator and will be at least as large as the key size chosen for the DEK. In this case, the external token authorization factor may be used directly as a submask in formation of the KEK.

1.1.3 Encryption

If the cryptography used to generate, handle, and protect keys or authorization factors is sufficiently robust and if the implementation has no critical mistakes, an adversary who obtains an unpowered lost or stolen hard drive without the authorization factors or KEK has to exhaust the encryption key space of the KEK or DEK to obtain the data. (Note that in cases where the passphrase offers less strength than the potential keyspace of the disk encryption algorithm, AES, and the passphrase is the only authorization factor unknown to the adversary, then the protection offered is commensurate with the passphrase strength rather than the keyspace of the AES key.)

Data on the hard disk are encrypted using the DEK. The DEK can be masked by either the KEK or an intermediate key. When an intermediate key is used, the intermediate key is masked by the KEK and the DEK is encrypted by the intermediate key. Any intermediate keys must meet the same strength requirements as the KEK and the DEK, and must be specified in the ST using iterated requirements.

The DEK is masked by the KEK (through an XOR operation or by using AES). The DEK will be generated using a Deterministic Random Bit Generator (DRBG) and will be either 128 or 256 bits. A properly seeded DRBG ensures a sample of noise at least equal to the key size of the DEK. The entropy used as input to the DRBG algorithm must be provided by at least one noise source, either a hardware-based source or a software-based source. It is important that the DEK and IV pair be unique on the system, and the level of granularity at which the cryptography is done (e.g., blocks, sectors, entire drive) is implementation dependent. So, if multiple drives or blocks on a drive are encrypted with the same DEK, different IV's should be used throughout the computer.

Unencrypted keys and keying material (including authorization factors) will be zeroized when no longer being used, and will not be written to persistent storage. No unencrypted keys or keying material should be available in plaintext form if an adversary recovers the hard disk device in its powered-down state.

1.1.4 Administration

As discussed further in 1.1.6, the base requirements of the TOE do not require the TOE to maintain an administrative role. However, the overall system will maintain the notion of an administrator of the TOE that is a subset of the users of the TOE. The term "administrator" is used in this sense in the following sections.

Administrators of the TOE shall correctly follow any required configuration guidance. The TOE can rely on the authentication system provided by the host O/S in the Operational environment to establish this role or can implement its own mechanisms, in which case information from Appendix C needs to be included in the Security Target. The TOE shall be capable of enforcing the following administrative functions:

- Create a DEK,
- Change an existing DEK,
- Change a passphrase-based authorization factor,
- Generating a Key Encryption Key (used to wrap the DEK) from a submask produced from an authorization factor.

Additional capability may be provided by the TOE and specified in the ST as long as the constraints of Appendix C are followed.

1.1.5 Authorized Users

Authorized users shall adhere to the user guidance to minimize the risk of compromised data. Authorization is determined by possessing and providing the TOE the correct authorization factor(s) to unlock the protected disk. It is the responsibility of the authorized users of the hard disk to secure and protect the device and the authorization factors for the TOE while it is in their possession. Authorized users must not let the hard drive leave their physical control while it is powered on. Authorized users shall not leave/store the passphrase and/or external token and/or TPM pin with the hard drive or if multiple factors are used, with each other. The external token should be removed from the system after successful authorization. The user will be provided appropriate guidance to maintain a secure TOE.

1.1.6 The TOE and Its Supporting Environment

The TOE supporting environment is significant. Since the TOE is purely a software solution, it must rely heavily on the TOE Operational Environment (system hardware, firmware, and operating system) for its execution domain and its proper usage. The vendor is expected to provide sufficient installation and configuration instructions to identify an Operational Environment with the necessary features and to provide instructions for how to configure it correctly.

The primary purpose of the TOE is to ensure data on the hard drive of the underlying platform are encrypted. However, it may be the case that some of the cryptographic services used by the TOE are provided by the Operational Environment. Cryptographic or key creation/manipulation requirements that are contained in the body of the document must be implemented by the TOE; for example, formation of the KEK from the authorization factors. Cryptographic or key creation/manipulation requirements that may be performed by either the TOE or the Operational Environment are contained in Appendix C. If the TOE performs these functions, the ST author moves these components to the body of the ST and makes the appropriate adjustments to the Threats and Objectives in the ST.

Section C.1 in Appendix C is somewhat different than other requirements in Appendix C for this PP. The requirements in Appendix C.1 **must** be met by either the TOE or the Operational Environment; in that sense they are not "optional" as are the other requirements in Appendix C. It is up to the evaluating Scheme to determine the evidence required from the developer/evaluation team to determine if the Operational Environment "meets" the requirements in C.1 that the TOE depends upon. The ST must enumerate the platforms that the TOE is to be evaluated on in sufficient detail so that a determination can be made as to whether that platforms can meet the requirements in C.1 that are allocated to it.

In some cases the TOE vendor will have to provide specific configuration guidance for the Operational Environment to enable the TOE to meet its security objectives. These include

- instructions for how to configure the power managed state (e.g., hibernate/sleep) capabilities so that the system powers down completely after a period of user inactivity for every operating system that the product supports;
- instructions for how to disable power managed state (e.g., hibernate/sleep) capabilities that are unable to be configured to power the system completely down;

- Instructions that describe how to disable any capability to use TOE authorization factors as part of or in place of Operational environment (operating system) identification and authentication information.

It should be noted that if the TOE possess the capability to correctly protect information in one or more of an underlying platform's power managed modes; they can use requirements in Appendix C.5.

Authorized users of the TOE are those users possessing valid authorization factors for the TOE. The TOE requires that certain management activities (defined in the FMT requirement) be performed by a subset of the authorized users of the TOE. This PP places no requirements on the TOE to provide an identification and authentication capability to restrict these management functions to an administrative role, which implies that there are a number of ways a TOE vendor could be compliant. For example,

1. The TOE contains no notion of an authorized administrator; anyone that can invoke the management utility can configure the TOE. In order to be compliant to the PP in this case, the TOE vendor must provide instructions as part of the AGD_OPE/PRE guidance that detail the procedures an administrator would use to configure the Operational environment such that only a subset of the authorized users of the TOE would be able to execute the management utility. For example, the guidance would describe configuring the access control mechanisms in the Operational environment so that only the administrator-allowed users would be able to execute the management utility. This case reflects the baseline requirements of this PP, as specified in A.PLATFORM_I&A.
2. The TOE contains a notion of an authorized administrator (or set of administrators), but relies on the Operational environment to perform the identification and authentication functions and then pass some indication to the TOE that can be matched to the internal TOE representation of an authorized administrator. In this case, the ST author will need augment the requirements (using the templates provided in Appendix C) to specify the capabilities provided by the TOE. The vendor will need to describe any configuration or settings in the Operational environment needed to support the passing of the information to the TOE.
3. The TOE contains its own identification and authentication capability that is used to determine which users of the system housing the hard disk are authorized to use the management functions provided by the TOE. In this case, the ST author will need to use the I&A template information provided in Appendix C in the body of the ST to specify this functionality.

2 Security Problem Definition

This Protection Profile (PP) is written to address the situation when a hard disk has been lost or stolen while powered off without prior access by an adversary.

2.1 Threats

A threat consists of a threat agent, an asset and an adverse action of that threat agent on that asset.

The threat agents are the entities that put the assets at risk if an adversary obtains a lost or stolen hard disk. For instance, a threat in the chart below is T.UNAUTHORIZED_DISK_ACCESS. The threat agent is the possessor (unauthorized user) of a lost or stolen hard disk. The asset is the data on the storage device, while the adverse action is to attempt to obtain those data from the hard disk. This threat drives the functional requirements for the disk encryptor (TOE) to authorize who can use the TOE to access the hard disk and encrypt/decrypt the data. Since possession of the KEK, DEK, intermediate keys, authorization factors, submasks, and random numbers or any other values that contribute to the creation of keys or authorization factors could allow an unauthorized user to defeat the encryption, keying material is considered equivalent to the data in importance and is the other asset addressed in the threat table.

It is important to reemphasize at this point that the product (TOE) is not expected in general to defend against the possessor of the lost or stolen hard disk who can introduce malicious code or exploitable hardware components into the Target of Evaluation (TOE) or the Operational Environment. It is assumed that the user will ensure the TOE is physically protected and that the Operational Environment provides sufficient protection against logical attacks. One specific area where some protection is offered by conformant TOEs is in providing updates to the TOE; other than this area, though, no countermeasures are mandated by this PP. Similarly, these requirements do not address the "lost and found" hard disk problem, where an adversary may have taken the hard disk, compromised the unencrypted portions of the boot device (e.g., MBR, boot partition), and then made it available to be recovered by the original user so that they would execute the compromised code.

Table 1: Threats

Threat	Description of Threat
T.KEYING_MATERIAL_ COMPROMISE	An attacker can obtain unencrypted key material (the KEK, the DEK, authorization factors, submasks, and random numbers or any other values from which a key is derived) that the TOE has written to persistent memory and use these values to gain access to user data.
T.PERSISTENT_INFORMATION	The TOE and/or 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_DISK_ACCESS	An unauthorized user that has access to the lost hard disk may gain access to data for which they are not authorized according to the TOE security policy.

T.UNAUTHORIZED_UPDATE	A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.
T.UNSAFE_AUTHFACTOR_VERIFICATION	An attacker can take advantage of an unsafe method for performing verification of an authorization factor, resulting in exposure of the KEK, DEK, or user data.

2.2 Assumptions

This section of the security problem definition shows the assumptions that are made on the operational environment in order to be able to provide security functionality. 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 anymore. Assumptions can be on physical, personnel and connectivity of the operational environment.

Table 2: TOE Assumptions

Assumption	Description of Assumption
A.AUTHORIZED_USER	Authorized users will follow all provided user guidance, including keeping passphrases and external tokens secure and stored separately from the disk.
A.ET_AUTH_USE_ONLY	External tokens that contain authorization factors will be used for no other purpose than to store the external token authorization factor.
A.PASSPHRASE_BASED_AUTH_FACTOR	An authorized administrator will be responsible for ensuring that the passphrase authorization factor has 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 shall remain unaffected by the TOE.
A.PROTECT_INTEGRITY	The user will exercise due diligence in physically protecting the TOE, and ensuring the IT environment will sufficiently protect against logical attacks.
A.SHUTDOWN	An authorized user 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_ADMINISTRATORS	Authorized administrators are appropriately trained and follow all administrator guidance.

3 Security Objectives

The Security Objectives are the requirements for the Target of Evaluation (TOE) and for the Operational Environment derived from the threats, organizational security policies, and the assumptions in Section 2. Section 4 restates the security objectives for the TOE more formally as Security Functionality Requirements (SFR). The TOE is evaluated against the SFRs.

3.1 Security Objectives for the TOE

Table 4 identifies the security objectives of the TOE. These security objectives reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. The TOE has to meet these objectives by satisfying the security functional requirements.

Table 3: Security Objectives for the TOE

Objective	Objective Description
O.AUTHORIZATION	The TOE must obtain the authorization factor(s) from a user to be able to decrypt the data on the hard disk.
O.CORRECT_TSF_OPERATION	The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF in its operational environment.
O.ENCRYPT_ALL	The TOE will encrypt all data that are stored on a hard drive. (Note that this may exclude the MBR and the bootable partition that it points to.)
O.DEK_SECURITY	The TOE will mask the DEK using a key encryption key (KEK) created from one or more submasks (which in turn are derived from the authorization factors) so that a threat agent who does not have authorization factor(s) will be unable to gain access to the user data by obtaining the DEK.
O.KEY_MATERIAL_COMPROMISE	The TOE will zeroize key material as soon as it is no longer needed to decrease the chance that such material could be used to discover a KEK or DEK.
O.MANAGE	The TOE will 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.
O.OWNERSHIP	The TOE shall ensure that ownership is taken (that is, a DEK is created, authorization factors are established, any default authorization factors are changed, a KEK is formed from the derived submasks, and the DEK is associated with the KEK) prior to any user data being accessible while the TOE is in operation.
O. SAFE_AUTHFACTOR_ VERIFICATION	The TOE shall perform verification of the authorization factors in such a way that the KEK, DEK, or user data are

	not inadvertently exposed.
O.TRUSTED_UPDATE	The TOE shall provide administrators the capability to update the TOE firmware/software, and verify that updates to the product are received from the intended source.

3.2 Security Objectives for the Operational Environment

The Operational Environment of the TOE implements technical and procedural measures to assist the TOE in correctly providing its security functionality (which is defined by the security objectives for the TOE). This part wise solution forms the security objectives for the Operational Environment and consists of a set of statements describing the goals that the Operational Environment should achieve.

This section defines the security objectives that are to be addressed by the IT domain or by non-technical or procedural means. The assumptions identified in Section 2.2 are incorporated as security objectives for the environment. They levy additional requirements on the environment, which are largely satisfied through procedural or administrative measures. Table 5 identifies the security objectives for the environment.

Table 4: Security Objectives for the operational environment

Objective	Objective Description
OE.PASSPHRASE_STRENGTH	An authorized administrator will be responsible for ensuring that the passphrase authorization factor conforms to guidance from the Enterprise using the TOE.
OE.PLATFORM_I&A	The Operational Environment will provide individual user identification and authentication mechanisms that operate independently of the authorization factors used by the TOE.
OE.POWER_SAVE[2]	The Operational environment must be configurable so that there exists at least one mechanism that will cause the system to power down after a period of time in the same fashion as the user electing to shutdown the system. Any such mechanism (e.g., sleep, hibernate) that does not conform to this requirement must be capable of being disabled by the administrator.
OE.RESTRICTED_FUNCTIONS	Management functions will be limited to an authorized administrator.
OE.SINGLE_USE_ET	External tokens that contain authorization factors will be used for no other purpose than to store the external token authorization factor.
OE.TRAINED_USERS	Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.

[2] If the TOE encompasses the platform and there is no "operational environment", then the ST author renames this objective "O.POWER_SAVE" and modifies the description to begin "The TOE must be configurable so that there…" The rationale will also need to be updated.

3.3 Security objective rationale

The security objective rationale for the TOE objectives is contained in Section 6. Table 6 illustrates the mapping from Security Objectives to Assumptions.

Table 6: Security Objectives to Assumptions Mappings

Assumption	Objectives Addressing the Assumption	Rationale
A.AUTHORIZED_USER Authorized users will follow all provided user guidance, including keeping passphrases and external tokens secure and stored separately from the disk.	OE.TRAINED_USERS Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.	OE.TRAINED_USERS ensures that users will be trained how to use the TOE correctly.
A.ET_AUTH_USE_ONLY External tokens that contain authorization factors will be used for no other purpose than to store the external token authorization factor.	OE.SINGLE_USE_ET External tokens that contain authorization factors will contain nothing other than the external token authorization factor. OE.TRAINED_USERS Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.	OE.SINGLE_USE_ET meets the objective by requiring that only the external token authorization factor(s) are resident on the external token. OE.TRAINED_USERS contributes by mandating that user will not put additional information on the external token.
A.PASSPHRASE_BASED_AUTH_FACTOR An authorized administrator will be responsible for ensuring that the passphrase authorization factor conforms to passphrase policy and has sufficient strength and entropy to reflect the sensitivity of the data being protected.	OE.TRAINED_USERS Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors. OE.PASSPHRASE_STRENGTH An authorized administrator will be responsible for ensuring that the passphrase authorization factor conforms to guidance from the Enterprise using the TOE.	OE.TRAINED_USERS satisfies this policy by ensuring that the administrator is trained and follows the guidance provided. OE.PASSPHRASE_STRENGTH satisfies this policy by ensuring that the administrator will create passphrases with 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	OE.PLATFORM_I&A The Operational Environment will provide individual user	OE.PLATFORM_I&A ensures that 1) the I&A mechanisms for users of the system are

Assumption	Objectives Addressing the Assumption	Rationale
identification and authentication. this I&A functionality shall remain unaffected by the TOE.	identification and authentication mechanisms that operate independently of the authorization factors used by the TOE.	implemented in the Operational environment, and 2) these mechanisms are not supplanted by the entry of authorization factors for the TOE (e.g., through a single sign-on capability).
A.PROTECT_INTEGRITY The user will exercise due diligence in physically protecting the TOE, and ensuring the IT environment will sufficiently protect against logical attacks.	OE.TRAINED_USERS Authorized users will be properly trained and follow all guidance for shutting down the TOE and securing authorization factors.	OE.TRAINED_USERS satisfies this assumption by providing the user with guidance on how to configure and operate the Operational Environment in a manner that will minimize exposure to logical attacks. Additionally, the guidance instructs the user to take measures to physically protect the TOE, even though the disk is encrypted.
A.SHUTDOWN An authorized user 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 Authorized users will be properly trained and follow all guidance for shutting down the TOE and securing authorization factors.	OE.TRAINED_USERS satisfies this assumption by instructing users how to power down the system correctly or ensuring the machine is in a power management mode that ensures sensitive information will not persist in non-volatile memory. The guidance also explains the importance of doing so.
A.STRONG_OE_CRYPTO All cryptography implemented in the Operational Environment and used by the TOE will meet the requirements listed in Appendix C of this PP. This includes generation of external token authorization factors by a RBG.	OE.STRONG_ENVIRONMENT_CRYPTO The Operating environment will provide a cryptographic function capability that is commensurate with the requirements and capabilities of the TOE and Appendix C.	It is acceptable for the TOE to make use of cryptographic functionality that is implemented in the Operational Environment. There are specific requirements for such cryptographic functionality, however, for example, characteristics of the RBG or the allowable modes for

Assumption	Objectives Addressing the Assumption	Rationale
		encryption using the DEK. Such specific requirements are captured in Appendix C and it is expected that the Operational Environment is able to provide the capabilities specified, thus satisfying this assumption.
A.TRAINED_ADMINISTRATORS Authorized administrators are appropriately trained and follow all administrator guidance.	OE.TRAINED_USERS Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.	OE.TRAINED_USERS ensures that administrators are trained and follow appropriate guidance to configure and maintain security of the TOE.

4 Security Requirements

The Security Requirements are divided into functional requirements and assurance requirements. The Security Functional Requirements (SFRs) are a formal instantiation of the Security Objectives and are provided with application notes in Section 4.1.

The Security Assurance Requirements (SARs) are typically inserted into a PP and listed separately from the SFRs; the CEM is then consulted during the evaluation based on the SARs chosen. Because of the nature of the Common Criteria Security Assurance Requirements and the specific technology identified as the TOE, a more tailored approach is taken in this PP. While the SARs are still listed for context and completeness in Section 4.3, the majority of the activities that an evaluator needs to perform for this TOE with respect to each SFR and SAR are detailed in "Assurance Activities" paragraphs. Assurance Activities are normative descriptions of activities that must take place in order for the evaluation to be complete. Assurance Activities are located in two places in this PP; those that are associated with specific SFRs are located in Section 4.1, while those that are independent of the SFRs are detailed in Section 4.3. Note that the Assurance Activities are in fact a tailored evaluation methodology, presented in-line for readability, comprehension, and convenience.

Future iterations of the Protection Profile may provide more detailed Assurance Activities based on lessons learned from actual product evaluations.

4.1 Security Functional Requirements

Table 5: TOE Security Functional Requirements

Functional Class	Functional Components
Cryptographic support Class (FCS)	FCS_CKM.1(1) Cryptographic key generation (DEK)
Cryptographic support Class (FCS)	FCS_CKM.1(2) Cryptographic key generation (KEK)

Cryptographic support Class (FCS)	FCS_CKM_EXT.4 Cryptographic keying material destruction
User data protection Class (FDP)	FDP_DSK_EXT.1 Extended: Protection of Data on Disk
Identification and authentication Class (FIA)	FIA_AUT_EXT.1 Extended: FDE User Authorization
Security management Class (FMT)	FMT_SMF.1 Specification of management functions
Protection of the TSF Class	FPT_TUD_EXT.1 Trusted Update
Protection of the TSF Class	FPT_TST_EXT.1 TSF Testing

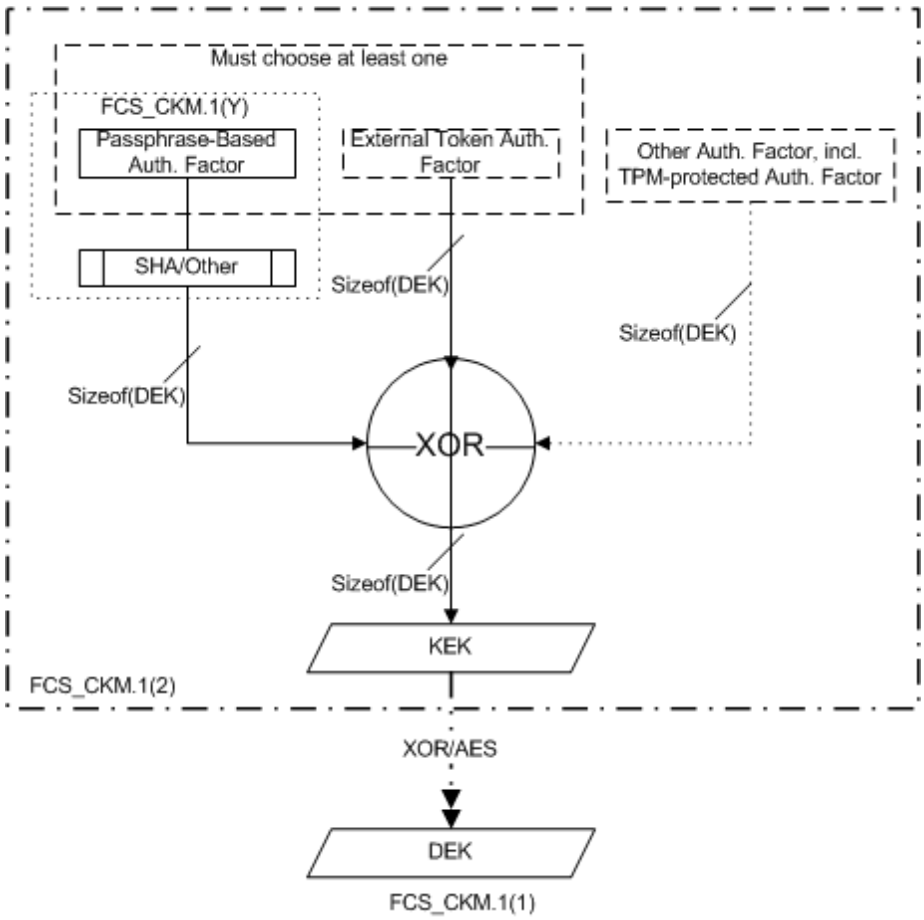
4.1.1 Class: Cryptographic Support (FCS)

The primary threats addressed by these functional requirements are a brute force attack against the key space and the failure of a cryptographic component.

The assurance requirements detail how the implementation of these requirements are to be verified. Each scheme has the option of specifying the process under which the cryptographic assurance activities may be considered satisfied.

Cryptographic Key Management (FCS_CKM)

Conformant implementations will contain at least two keys: a Key-Encrypting Key (KEK) and a Data Encryption Key (DEK). The following requirements specify how the keys are produced. Because this production is slightly complicated, the following illustration provides an overview of the requirements in this section.



As pictured above, generation of the DEK is specified in FCS_CKM.1(1), while the KEK generation is specified in FCS_CKM.1(2) using passphrases and/or external token authorization factors. Passphrases must be conditioned as specified in FCS_CKM.1(Y) in section C.4 Authorization Factors. The KEK is generated from submasks derived from authorization factors. A TOE is required to support either a passphrase-based authorization factor (FCS_CKM.1(Y)) or an external token containing an authorization factor; it can also support both of these. A note on external authorization factors: the TOE is not required to generate such factors, but can use them without generating them. Since FCS_CKM deals with the generation of keys, if the TOE does want to claim credit for the ability to generate external token authorization factors as well, then the FCS_CKM.1(X) component (and associated threats, objectives, and rationale) from Appendix C must be included in the body of the ST.

As long as the TOE supports at least one of these authorization factors, it can also support other authorization factors, including or a TPM-based authorization factor (shown on the right of the diagram above); these are specified in FCS_CKM.1(2). FCS_CKM.1(2) also specifies how the various authorization factors are combined to form the KEK. With this background, the key generation requirements are presented below.

FCS_CKM.1(1) Cryptographic key generation (DEK)

FCS_CKM.1.1(1) **Refinement:** The TSF shall generate **DEK** cryptographic keys **using a Random Bit Generator as specified in FCS_RBG_EXT.1 and** specified cryptographic key sizes **[selection: 128 bit, 256 bit]** that meet the following: [**No Standard]**.

Application Note: The intent of this requirement is to ensure that the DEK cannot be recovered with less work than a full exhaust of the key space for AES. The key generation capability of the TOE uses a RBG implemented on the TOE device. Either 128-bit or 256-bit (or both) are allowed; the ST author makes the selection appropriate for the device. A DEK is used in addition to the KEK so that authorization factors (especially the passphrase authorization factor) can be changed without having to re-encrypt all of the user data on the device.

FCS_RBG_EXT.1 is contained in Appendix C; it is allowable for the DEK generation routine in the TOE to call out to supporting cryptographic functions such as a random bit generator and still be conformant to this PP.

If the TOE uses intermediate keys this requirement will be iterated for intermediate keys.

Assurance Activity: The evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked. If the RBG is provided by the Operational Environment, then the evaluator checks to ensure that--for each platform identified in the ST--the TSS describes the interface used by the TOE to invoke this functionality. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT or documentation available for the operational environment to determine that the key size being requested is identical to the key size and mode to be used for the encryption/decryption of the user data (FCS_COP.1(1), which is also a requirement contained in Appendix C).

FCS_CKM.1(2) Cryptographic key generation (KEK)

FCS_CKM.1.1(2) **Refinement:** The TSF shall **derive KEK** cryptographic keys in accordance with a specified cryptographic key **derivation** algorithm **[selection: None, exclusive OR (XOR)] with the following inputs [selection:**

a submask derived from a passphrase authorization factor composed and conditioned as defined in FCS_CKM.1(Y),

an external token authorization factor that is at least the same bit-length as the DEK,

[selection: no other inputs, a TPM-protected authorization factor that is at least the same bit-length as the DEK, [assignment: list of other authorization factors and associated submask derivation methods] that produce submasks that are at least the same size as the DEK as specified in FCS_CKM.1(1)]]

maintaining the effective strength of each authorization factor and specified

cryptographic key sizes **[selection: 128 bits, 256 bits]** that meet the following: **[No standard].**

Application Note: These requirements are intended to define how the authorization factors are used to create the KEK. While specific guidance to the ST author is provided below for each assignment and selection, the following is a high-level description of the point of this component. The ST author chooses a passphrase authorization factor, and/or an external token authorization factor. The ST Author then has the option of choosing a TPM-protected authorization factor or defining additional authorization factors. If any additional authorization factors are defined, then the method by which submasks are produced from these authorization factors must also be described. The only condition levied on such an assignment is that the submasks produced be at least the same size as the DEK, and if cryptographic operations are used to form the submasks, those cryptographic operations are specified using additional FCS_COP.1 iterations. Use of multiple authorization factors is preferable; if more than one authorization factor is used, the submasks produced must be combined using XOR.

For the first selection, the ST author chooses "None" if only one authorization factor is used. If more than one authorization factor is used, then the ST author chooses "XOR" (no other combining methods are conformant with this PP).

For the second selection, the ST author selects the authorization factors used. Note that more than one can be selected, but at least one of the two (passphrase-based, external token-based) must be selected. If additional authorization factors beyond a conditioned passphrase or those external to the TOE are used, then the ST author uses the assignment within the selection within this second selection to specify these factors (or selects "no other inputs"). If the passphrase-based selection is chosen, the ST author includes FCS_CKM.1(Y) from Appendix C in the body of the ST. For TPM-protected and external authorization factors, the authorization factor contained on the external device/TPM is not conditioned, but rather is used directly.

For the cryptographic key size selection, the size of the KEK that is produced is chosen; this must be the same bit length as was specified for the DEK in FCS_CKM.1(1).

Note that "No Standard" is needed because either there is only one authorization factor used to form the KEK (whose composition is specified elsewhere), or the KEK is formed using the XOR function.

Assurance Activity: The assurance activity for this component entails examination of the ST's TSS to determine that the TOE's implementation of the requirements is documented. The evaluators shall first examine the TSS section to ensure that the authorization factors specified in the ST are described. For passphrase-based factors the examination of the TSS section is performed as part of FCS_CKM.1(Y) assurance activities. For external authorization factors, the TSS shall detail

whether the authorization factor must be generated by the TOE (in which case the assurance activities associated with FCS_CKM.1(X) in Appendix C apply); if not, then the TSS section shall specify measures taken by the TOE (or administrative personnel) to ensure the external authorization factor meets the minimum length requirements listed in the ST. Acceptable means include administrator-verification of the length or input validation checks performed at run time by the TOE. Additionally in this case, the evaluator shall verify that the administrative guidance discusses the characteristics of external authorization factors (e.g., how the authorization factor must be generated; format(s) or standards that the authorization factor must meet, configuration of the TPM device used) that are able to be used by the TOE.

If other authorization factors are specified, then for each factor, the TSS specifies how the factors are input into the TOE; how a submask is produced from the authorization factor (including any associated standards to which this process might conform), and verification performed to ensure the length of the submask meets the required size (as specified in this requirement).

If there is only one authorization factor, it naturally is not combined with anything and so no assurance activity is associated with this case. If the submasks produced from the authorization factors are XORed together to form the KEK, the TSS section shall identify how this is performed (e.g., if there are ordering requirements, checks performed, etc.). The evaluator shall also confirm that the TSS describes how the length of the output produced is at least the same as that of the DEK.

The evaluator shall also perform the following tests:

- Test 1 [conditional]: If there is more than one authorization factor, ensure that failure to supply a required authorization factor does not result in access to the encrypted data.
- Test 2 [conditional]: If the TOE supports multiple external tokens of different formats, the evaluators confirm that each format is able to be successfully used in providing authorization information to the TOE.

FCS_CKM_EXT.4 Cryptographic Key Material Destruction

FCS_CKM_EXT.4.1 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

Application Note: "Cryptographic Critical Security Parameters" are defined in FIPS 140-2 as "security-related information (e.g., secret and private cryptographic keys, and authentication data such as passwords and PINs) whose disclosure or modification can compromise the security of a cryptographic module."

The zeroization indicated above applies to each intermediate storage area for plaintext key/cryptographic critical security parameter (i.e., any storage, such as

memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another location.

Assurance Activity: The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and CSPs used to generate key; when they are zeroized (for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the zeroization procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are zeroized by overwriting once with zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write").

4.1.2 Class: User Data Protection (FDP)

This family stipulates encryption of all stored data.

Extended: Protection of Data on Disk (FDP_DSK_EXT)

FDP_DSK_EXT.1	**Extended: Protection of Data on Disk**
FDP_DSK_EXT.1.1	The TSF shall perform Full Disk Encryption in accordance with FCS_COP.1.1(1).
FDP_DSK_EXT.1.2	The DEK can only exist on an unpowered hard disk if it is masked with a KEK (or intermediate key) derived as specified in FCS_CKM.1(2) and FCS_COP.1(4).
FDP_DSK_EXT.1.3	The TSF shall encrypt all data without user intervention.
FDP_DSK_EXT.1.4	No plaintext keying material shall be written to persistent storage.

Application Note: “Full Disk Encryption” is defined in the Glossary for this PP as “the process of encrypting all the data on the hard drive of a computer, including the computer’s OS, and permitting access to the data only after successful authentication to the FDE product” with the exception of the MBR and associated bootable partition containing the code necessary to accept and process the authorization factors.

The intent of this requirement is to specify that encryption of any critical file will not depend on a user electing to protect that file. The disk encryption specified in FDP_DSK_EXT.1 occurs transparently to the user and the decision to protect the data is outside the discretion of the user, which is a characteristic that distinguishes it from file encryption.

This requirement is addressing not just explicitly stored files containing user data but includes chunks of data that are stored in swap files, registries, and other Operational Environment storage areas on the disk. It does not require encryption of all removable media. However, it does require encryption of all hard drives.

It should be noted that the intent of FDP_DSK_EXT.1.4 is that the TOE will not write plaintext keying material to persistent storage. In most cases the TOE cannot control keying material in memory that is written to swap space on the disk, for example. Since these areas of the disk are encrypted with the DEK, this exception to FDP_DSK_EXT.1.4 is acceptable.

Assurance Activities: The evaluator shall consult the TSS section of the ST in performing the assurance activities for this requirement. The evaluator focuses on ensuring that the description is comprehensive in how the data are written to the disk and the point at which the encryption function is applied. Since the implementation of the encryption/decryption functionality is implemented entirely in software on the host operating system, then the TSS must make the case that all methods of accessing the disk will pass through these functions. This must be true for all hard drives contained on the platform on which the TOE runs.

In performing their review, the evaluator shall determine that the TSS contains a description of the activities that happen on power-up relating to the loading of the MBR and the portions of the TOE that perform the authorization function. The TSS should also cover the initialization of the TOE and the activities that are performed to ensure that all disks are entirely encrypted when the TOE is first established; areas of the disk that are not encrypted (e.g., portions associated with the MBR) shall also be described.

The evaluator shall ensure that the description also covers how the cryptographic functions in the FCS requirements are being used to perform the encryption functions, including how the DEK is unwrapped and stored in the TOE. For the cryptographic functions that are provided by the Operational Environment, the evaluator shall check the TSS to ensure it describes--for each platform identified in the ST--the interface(s) used by the TOE to invoke this functionality. The evaluator shall ensure that the TSS describes, for each power-down scenario (see assurance activities for FCS_CKM_EXT.4) how the TOE ensures the DEK is wrapped with the KEK.

The evaluator shall ensure that the TSS describes how other functions available in the system (e.g., regeneration of the DEK) ensure that no unencrypted data or key material are present on the disk.

If the TOE supports multiple disk encryption, the evaluator shall examine the administration guidance to ensure the initialization procedure addresses the need for all disks on the platform to be encrypted.

The evaluator reviews the TSS to determine that it makes a case that key material is not written unencrypted to the hard disk. Since in normal use all writes to the disks will be encrypted, one approach is to make an argument concerning the exceptional cases when data are written to the unencrypted portions of the disk, then detailing how key material is prevented from being written in these areas.

The evaluator shall review the AGD guidance to determine that it describes the initial steps needed to enable the FDE function, including any necessary preparatory steps. The guidance shall provide instructions that are sufficient, on all platforms, to ensure that all hard drive devices will be encrypted when encryption is enabled.

The evaluators shall perform the following test activities:

- Test 1: Ensure that following the initialization activities results in all disks being encrypted. For areas of the device(s) that are found to be unencrypted, ensure justification is provided (in the TSS or operational guidance, for instance) that no user data or sensitive TSF data can be written to these areas. The examination of the disks can be done in several ways. Physically removing the drive and then inserting it into another computer is one method. Alternatively, booting the system that contains the encrypted hard drive from an external device and then directly accessing the encrypted drive is another example of an acceptable method for examination.
- Test 2: Ensure that the data (including data that may be stored in page files in the OS) are encrypted when written to disk. The extent that this is tested is consistent with the previous test; that is, it is acceptable to power on the system "normally", cause data to be written to the disk, and then use the methods mentioned in the previous test to ensure those data do not appear unencrypted on the device(s).

4.1.3 Class: Identification and Authentication (FIA)

Extended: FDE User Authorization (FIA_AUT_EXT)

FIA_AUT_EXT.1	**Extended: FDE User Authorization**
FIA_AUT_EXT.1.1	The TSF shall use a mechanism as defined in FCS_CKM.1.1(2) and FCS_COP.1(4) to perform user authorization.
FIA_AUT_EXT.1.2	The TSF shall perform user authorization using the mechanism provided in FIA_AUT_EXT.1.1 before allowing access to user data from the device.
FIA_AUT_EXT.1.3	The TSF shall verify that the authorization factors are valid before allowing

access to unencrypted data from the device.

FIA_AUT_EXT.1.4 The TSF shall ensure that the method of validation for each authorization factor does not expose or reduce the effective strength of the KEK, DEK, or CSPs used to derive the KEK or DEK.

Application Note: The intent of this requirement is to specify the mechanisms by which users are authorized to decrypt the disk and thus gain access to their system. Note that this is not considered authentication of an individual user. The authorization factors can be cloned and provided to all authorized users of a hard disk. Or, the user could have an authorization factor that is unique to the user.

Future versions of this PP may require that when an external token authorization factor is used, the booting process will stop and not continue until the user has removed the token.

A vendor can create intermediate keys that are essentially chains of KEKs. Any key that is encrypted by another key must meet the requirements for a DEK encrypted by a KEK. Every intermediate key has to be encrypted by another key. The ST author should capture this in the ST by iteration of the FCS_CKM and FCS_COP requirements.

If the authorization factors for the hard disk are lost, then data can only be recovered if the DEK was exported from the TOE or if the authorization factors were backed up (or if the DEK was encrypted by a different set of authorization factors, which were not lost). If the encrypted DEK is corrupted in the TOE, then a backup of the authorization factors will not be sufficient to restore the data.

Elements 1.3 and 1.4 deal with the validation of the authorization factors provided by the user prior to a user being able to access the information on the device. If the authorization factor is not valid, it is undesirable for the TSF to attempt to form a KEK, use it to unmask the DEK, and then present gibberish to the user. However, checking that the authorization factor is valid should not be done in a way that allows an attacker to circumvent the other requirements; since this operation is typically done on the host, it may be monitored/disassembled by an attacker and so must be designed with this threat in mind.

User authorization only needs to be performed when the device is made accessible to the user (that is, at system boot time). The above requirement should not be interpreted to mean that user authorization has to take place prior to every device or file access.

Assurance Activity: The evaluator shall check the TSS section to determine it describes how the TOE is initialized; that is, the sequence of events including power on; MBR access; and load of the code that will perform the authorization activities. If the operational guidance describes different start-up modes (e.g., pressing certain

function keys during the boot process), the evaluator shall ensure the TSS describes how these modes do not allow access the data on the hard disk prior to entering the authorization factors.

The evaluator shall check that the TSS describes how the authorization factors are validated prior to allowing the user to access the data on a drive. This description shall be in enough detail so that the evaluator can determine that the method or methods used do not expose the DEK, KEK, or other key material. "Expose" also includes the notion of weakening the DEK or KEK. It is not required to have a separate method for checking each authorization factor if separate authorization factors are used to provide submasks to create the KEK. The evaluator shall document their analysis of the mechanism(s) used to authenticate the authorization factors in the test report (ATE_IND).

For the cryptographic functions implemented in the Operational Environment that are used by the TOE in implementing this component, the evaluator shall check the TSS to ensure it describes--for each platform identified in the ST--the interface(s) used by the TOE to invoke this functionality.

The evaluator shall perform the following tests:

- Test 1: Ensure that the authorization factors are prompted for prior to allowing any access to unencrypted data on the hard drive devices.

- Test 2: For each supported authorization factor, ensure that incorrect entry of an authorization factor results in a notification from the TOE that an incorrect authorization has been provided.

- Test 3 [conditional]: If any bypass or alternate boot modes are provided, test to ensure that the modes are consistent with the requirements (that is, the appropriate authorization factors have to be entered prior to access to unencrypted data).

- Test 4: Ensure that after access to the device is obtained by correctly entering the authorization factors, the underlying platform still requires identification and authentication distinct from the authorization factors already entered.

4.1.4 Class: Security Management (FMT)

The primary intent in this section is to call out critical activities that must be (or must not able to be) performed by an administrator to prevent a negligent user from putting the disk encryptor in an insecure state. The administration model for conformant TOEs is described in Section 1.1.4 of this PP. If additional capabilities are provided by the TOE, the appropriate management and I&A requirements from Appendix C should be included in the ST.

Specification of Management Functions (FMT_SMF)

FMT_SMF.1 **Specification of Management Functions**

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

a) **generate the DEK when the disk drive is initialized for encrypted operation or on command by the administrator**
b) **Wrap the DEK using a KEK formed from submasks derived from user-entered authorization factors; specifically a [selection, one or more of: passphrase-based authorization factor, external token authorization factor] and [selection: No other factors, TPM-protected authorization factor, [assignment: other authorization factor]]**
c) **[selection: choose one of: no other functions, [selection: change passphrase-based authorization factor, change default authorization factors, generate external authorization factors, configure cryptographic functionality, disable key recovery functionality, [assignment: other management functions provided by the TSF]].]**

Application Note: The intent of this requirement is to express the management capabilities that the TOE possesses. This means that the TOE must be able to perform the listed functions. Items (a) and (b) establish the necessary keying material for operational use, and Item (c) is used to specify functionality that may be included in the TOE, but is not required to conform to the PP.

For Item b, the appropriate authorization factors that contribute to the formation of the KEK as per FCS_CKM.1(2) should be specified by the ST author. From an implementation point of view, this binds the authorization factors to the DEK so that the factors can be provided to end users. These authorization factors must be entered or generated under user control; while key recovery capability may exist, it must be disabled (or have the ability to be disabled) such that there are no recovery keys generated when the DEK/KEK are generated. If "passphrase-based authorization factor" is chosen in item b, then the ST author also chooses "change passphrase-based authorization factor" in item c.

In item d, if no other management functions are provided (or claimed), then “no other functions” should be selected. Several other common options are given:

- If “generate external authorization factors” is included, then the requirement in Appendix C, C.4 must be included in the ST.
- If the TOE provides configurability of the cryptographic functions (for example, key size of the DEK), then “configure cryptographic functionality” will be included, and the specifics of the functionality

offered can either be written in this requirement as bullet points, or included in the TSS.

- If the TOE does include a key recovery function, the TOE must provide the capability for the user to turn this functionality off so that no recovery key is generated.
- If "other management functions" are assigned, the National Scheme overseeing the evaluation must be consulted to ensure the assurance activities and other functionality requirements that may be needed are appropriately specified so that the ST can claim conformance to this PP.

Assurance Activity: The assurance activities for this component will be driven by the selections made by the ST author. This section describes assurance activities for the selections in the above component (with the exception of the "other management functions" assignment); it should be understood that if a capability is not selected in the ST, the noted assurance activity does not need to be performed. The following sections are divided up into "Required Activities" and "Conditional Activities" for ease of reference.

Required Activities

While it is a requirement of products conformant to this PP to restrict (with potentially significant help from the Operational Environment) the above-mentioned functions to an administrator (a privileged group that is a subset of the set of users of the TOE), the requirement to do so and associated assurance activities are levied elsewhere in the ST, depending on the particular TOE implementation. As detailed in section 1.1.4 of this PP, there are a range of scenarios that could be implemented, and the assurance activities associated with each—while similar—differ in scope and implementation.

General
The evaluator reviews the TSS and AGD guidance and shall determine that it identifies all of the non-TOE products needed to perform administration. For instance, if the management capability is provided via Java Applets or through a web browser interface, the details of what needs to be in place are provided in the TSS.

Generate DEK
The evaluator reviews the AGD guidance and shall determine that the instructions for generating a DEK exist. The instructions must cover all environments on which the TOE is claiming conformance, and include any preconditions that must exist in order to successfully generate or re-generate the DEK. The TSS is checked to ensure that the description of how the DEK is generated is consistent with the instructions in the AGD guidance, and any differences that arise from different platforms are taken into account. The TSS shall also describe the processing (if any) that occurs for existing data when a new DEK is generated and installed. The evaluator shall also perform the following tests:

- Test 1: On a “clean” installation, the administrator is able to generate the DEK.
- Test 2: With the disk already encrypted, generate a new DEK for the disk(s) and verify that the new DEK is different from the previous DEK.
- Test 3: Using information in the TSS and AGD_OPR/AGD_PRE guidance, ensure any claims made relating to functionality that is visible to users of the TOE as a function of the DEK regeneration process are verified (e.g., files encrypted with the previous DEK are still visible after a new DEK is generated).

Protect the DEK with a KEK formed from submasks from appropriate authorization factors

The ST will specify the authorization factors that are supported by the TOE, and provide requirements on how many factors—and in what combinations—are necessary to successfully use the TOE functions (this is done in the FCS_CKM requirements). This requirement deals with the initial wrapping of the DEK (or the wrapping of a new DEK) with the KEK produced from the authorization factors. The authorization factors can be per-disk or per-user. The evaluator shall review the AGD guidance to determine, for each supported authorization factor, the guidance details how that factor is input into the TSF for this operation. The evaluator shall review the TSS section to determine that it describes how the various authorization factors are combined to form the KEK, and how the KEK is used to mask the DEK; it shall also be clear whether there are any differences between this process and the process used during “normal” operations (that is, once the TOE is established) . This description could also encompass the information described in the assurance activities for the FCS_CKM.1* requirements. If the supported authorization factors are different depending on the platform, then the AGD guidance shall be clear on the minimum requirements for each platform, and any other limitations concerning authorization factors that apply. The evaluator shall also perform the following tests:

- Test 4: For each acceptable combination of authorization factors, establish a DEK and then ensure that the administrator is able to enter the authorization factors that result in the DEK being encrypted.

  The number of different times this test is performed is dependent on the number of authorization factors allowed by the implementation. Authorization factors should be tested on a variety supported Operational Environments (as claimed in the TOE), especially if there are different authorization factors supported in different environments (for example, environments possessing a TPM vs. those that don’t).

Conditional Activities

Item c in the above requirement contains several selections specifying functionality that may be provided by the TOE, but is not required to be conformant to this PP. If the functionality is provided, though, the TOE can claim conformance by including the appropriate requirements from Appendix C and making the corresponding selection above. As noted in the application note, if an assignment is made, the Nation Scheme overseeing the evaluation needs to be consulted to determine if compliance to the PP can be claimed.

If passphrase-based authorization factors are used by the TOE, then the "change passphrase-based authorization factor" item is selected and the following assurance activities shall be performed. The evaluator shall examine the operational guidance to ensure that it describes how the passphrase-based authorization factor is to be changed. The evaluator shall also examine the TSS to ensure that it describes the sequence of activities that take place on the host and on the hard disk device when this activity is performed, and ensure that the KEK and DEK are not exposed during this change. The evaluator shall also perform the following test:

- Test 5 [conditional]: The evaluator shall establish a passphrase authorization factor for the hard disk device. The evaluators shall then transfer user data from the host to the device. They shall then use the "change authorization factor" functionality to change the passphrase on the device. Once the current authorization factor is changed, they shall ensure that they are still able to access the data on the device. The evaluator shall also use the old authorization factor (after successfully changing the authorization factor) to show that it no longer provides access to the user data on the device.

It may be the case that the hard disk device arrives with default authorization factors in place. If it does, then the selection in section d must be made so that there is a mechanism to change these authorization factors. The operational guidance shall describe the method by which the user changes these factors when they are taking ownership of the device. The TSS shall describe the default authorization factors that exist. The evaluators also perform the following test:

- Test 6: [conditional] If the TOE provides default authorization factors, the evaluator shall change these factors in the course of taking ownership of the device as described in the operational guidance. The evaluator shall then confirm that the (old) authorization factors are no longer valid for data access.

Two of the activities in section d concern the generation of authorization factors: passphrases and the bit-string to be put on an external token. In each of these cases, additional requirements from Appendix C will be included in the ST; associated with these requirements are the assurance activities covering the details of how the authorization factors are generated. For this requirement,

the evaluator shall review the AGD information to ensure that the instructions for invoking the authorization factor mechanism are detailed and clear enough so that an authorization factor with the required characteristics can be generated. The tests associated with these mechanisms are specified as part of that particular mechanisms' assurance activities.

In some TOEs, there may be a choice with respect to the underlying cryptography that is used; for instance, the length of the DEK in bits, or the encryption mode that is used for AES. Again, this capability does not have to be offered for the TOE to claim conformance to the PP; however, if the capability is offered, it is specified in the ST and the "configure cryptographic functionality" choice is selected in the requirement above.

For this selection, the evaluator shall determine from the ST what portions of the cryptographic functionality are configurable. This will entail looking at the FCS requirements as well as the associated description in the TSS, as well as an additional documentation associated with the TOE in terms of the cryptographic functionality. Armed with this information, the evaluator shall review the AGD documentation to determine that there are instructions for manipulating all of the claimed mechanisms. The evaluator shall also perform the following test:

- Test 7 [conditional]: For each configurable cryptographic mode supported, the evaluator follows the AGD instructions to determine that TOE functionality operates as expected (that is, the hard disk(s) are encrypted/decrypted appropriately). While it is not required that the details be verified at this assurance level (that is, if the TOE allows AES 128-bit or 256-bit keys, the evaluator is not required to run the process in a debugger that allows them to determine the length of the key), it is required that some comparative analysis be performed. For example, if different AES modes are supported, choosing a different mode with the same DEK should result in different ciphertext being present on the hard disk(s).

If the TOE supports key recovery, this must be stated in the TSS. The TSS shall also describe how to disable this functionality. This includes a description of how the recovery material is provided to the recovery holder. The intent is that this description can be used by the evaluators in testing to determine whether the key recovery functionality has indeed been disabled (for instance, if the TSS states that the material is sent to the third party through a network connection when a new KEK/DEK is generated, the evaluators can disable the functionality, attach a network monitor, and see whether a network connection is made when a new KEK/DEK is generated). The guidance for disabling this capability shall be described in the AGD documentation.

- Test 8 [conditional] If the TOE provides key recovery capability whose effects are visible at the TOE interface, then the evaluator shall devise a

test that ensures that the key recovery capability has been or can be disabled following the guidance provided by the vendor.

4.1.5 Class: Protection of the TSF (FPT)

Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1	**Extended: Trusted Update**
FPT_TUD_EXT.1.1	The TSF shall provide administrators the ability to query the current version of the TOE firmware/software.
FPT_TUD_EXT.1.2	The TSF shall provide administrators the ability to initiate updates to TOE software.
FPT_TUD_EXT.1.3	The TSF shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [selection: published hash, no other functions] prior to installing those updates.

Application Note: The digital signature mechanism referenced in the third element is the one specified in FCS_COP.1(2) in Appendix C. The published hash referenced is generated by one of the functions specified in FCS_COP.1(3) in Appendix C. While this component requires the TOE to implement the update functionality itself, it is acceptable to perform the cryptographic checks using functionality available in the Operational Environment.

Assurance Activities: Updates to the TSF are signed by an authorized source, and may have an associated hash. The definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are installed in the Operational Environment. The evaluator ensures this information is contained in the TSS and that any instructions dealing with the installation of the update credentials is detailed in the operational guidance. The evaluator also ensures that the TSS (or the operational guidance) describes how the candidate updates are obtained; the processing associated with verifying the digital signature of the updates (and optionally calculation of the hash); and the actions that take place for successful (signature was verified and optionally hash was verified) and unsuccessful (signature could not be verified, optional hash was incorrect) cases. If the digital signature/hashing is performed by the Operational Environment, then the evaluator shall check the TSS to ensure it describes--for each platform identified in the ST--the interface(s) used by the TOE to invoke this cryptographic functionality.

The location of the software that is performing the processing must also be described in the TSS and verified by the evaluators. The evaluators shall perform the following tests (if an optional hash is supported by the TOE, then

the evaluator performs tests 2 and 3 for different combinations of valid and invalid digital signatures and hashes, as well as for digital signature alone):

- Test 1: The evaluator performs the version verification activity to determine the current version of the product. After the update tests described in the following tests, the evaluator performs this activity again to verify that the version correctly corresponds to that of the update.
- Test 2: The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Perform a subset of other assurance activity tests to demonstrate that the update functions as expected.
- Test 3: The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. The evaluator verifies that the TOE rejects the update.

Extended: TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1	**Extended: TSF Testing**
FPT_TST_EXT.1.1	The TSF shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.
Assurance Activities:	If FCS_RBG_EXT.1 is implemented by the TOE and according to NIST SP 800-90, the evaluator shall verify that the TSS describes health tests that are consistent with section 11.3 of NIST SP 800-90. If any FCS_COP functions are implemented by the TOE, the TSS shall describe the known-answer self-tests for those functions. The evaluator shall verify that the TSS describes, for some set of non-cryptographic functions affecting the correct operation of the TSF, the method by which those functions are tested. The TSS will describe, for each of these functions, the method by which correct operation of the function/component is verified. The evaluator shall determine that all of the identified functions/components are adequately tested on start-up.

4.2 Security Assurance Requirements

The Security Objectives for the TOE in Section 3.1 were constructed to address threats identified in Section 2.1. The Security Functional Requirements (SFRs) in Section 4.1 are a formal instantiation of the Security Objectives.

As indicated in the introduction to Section 4.1, while this section contains the complete set of SARs from the CC, the Assurance Activities to be performed by an evaluator are detailed both in section 4.1 as well as in this section.

For each family, “Developer Notes” are provided on the developer action elements to clarify what, if any, additional documentation/activity needs to be provided by the developer. For the content/presentation and evaluator activity elements, additional assurance activities (to those already contained in section 4.1) are described as a whole for the family, rather than for each element. Additionally, the assurance activities described in this section are complementary to those specified in section 4.1.

The TOE security assurance requirements, summarized in Table 10, identify the management and evaluative activities required to address the threats and policies identified in Section 3 of this PP. Section 4.3 provides a succinct justification for choosing the security assurance requirements in this section.

Table 6: TOE Security Assurance Requirements

Assurance Class	Assurance Components	Assurance Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
Tests	ATE_IND.1	Independent testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability analysis
Life Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage

4.2.1 Class ADV: Development

For TOEs conforming to this PP, the information about the TOE is contained in the guidance documentation available to the end user as well as the TOE Summary Specification (TSS) portion of the ST. While it is not required that the TOE developer write the TSS, the TOE developer must concur with the description of the product that is contained in the TSS as it relates to the functional requirements. The Assurance Activities contained in Section 4.1 should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

4.2.1.1 ADV_FSP.1 Basic functional specification

The functional specification describes the TSFIs. At the level of assurance provided by this PP, it is not necessary to have a formal or complete specification of these interfaces. Additionally, because TOEs conforming to this PP will necessarily have interfaces to the Operational Environment that are not directly invokable by TOE users (to include administrative users), at this assurance level there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. The activities for this family for this PP should focus on understanding the interfaces presented in the TSS in response to the functional requirements, and the interfaces presented in the AGD documentation. No additional "functional specification" document should be necessary to satisfy the assurance activities specified.

In understanding the interfaces to the TOE, it is important to consider that the threat that is to be countered is that the attacker finds an unpowered hard disk and wishes to get at the data on the disk. Since this means that the attacker does not have access to the system when the TOE is operational (that is, the authentication factors have been successfully input and the TOE is encrypting and decrypting data being transferred between the system and disk), the primary untrusted user interface is the one presented to the user when the system is booted. In addition to these "user" interfaces, the administrative interface (how the TOE is configured) also needs to be described.

In some cases, there might be other interfaces that can be directly invoked such as the interface on a hard disk (for instance, removing the hard drive and putting it on another computer as a secondary device), but it is not necessary in these cases for the developer to deliver this type of interface specification or for evaluators to examine the entire USB or SCSI interface implementation, for example, looking for errors. The interfaces that need to be evaluated are characterized through the information needed to perform the assurance activities listed, rather than as an independent, abstract list.

Developer action elements:

ADV_FSP.1.1D	The developer shall provide a functional specification.
ADV_FSP.1.2D	The developer shall provide a tracing from the functional specification to the SFRs.
Developer Note:	As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPR and AGD_PRE documentation, coupled with the information provided in the TSS of the ST. The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

Content and presentation elements:

ADV_FSP.1.1C	The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.2C	The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.
ADV_FSP.1.3C	The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.
ADV_FSP.1.4C	The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

ADV_ FSP.1.1E	The evaluator *shall confirm* that the information provided meets all requirements for content and presentation of evidence.
ADV_ FSP.1.2E	The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

Assurance Activities:

There are no specific assurance activities associated with this component. The interface documentation provided to support the evaluation activities is described in section 4.1, and other activities are described for AGD, ATE, and AVA SARs.

4.2.2 Class AGD: Guidance Documents

The guidance documents will be provided with the developer's security target. Guidance must include a description of how the administrator verifies that the Operational Environment (the product that hosts the hard disks) can fulfill its role for the security functionality. The documentation should be in an informal style and readable by an administrator.

Guidance must be provided for every Operational Environment that the product supports as claimed in the ST. This guidance includes

- instructions to successfully install the TOE in that environment; and
- instructions to manage the security of the TOE as a product and as a component of the larger Operational environment.

Guidance pertaining to particular security functionality is also provided; requirements on such guidance are contained in the assurance activities specified in section 4.1.

In addition to the areas already mentioned, the guidance specifies which power management modes (e.g., hibernate, sleep) conform to OE.POWER_SAVE and provides instructions how to disable those that do not conform to be disabled.

4.2.2.1 AGD_OPE.1 Operational User Guidance

Developer action elements:

AGD_OPE.1.1D The developer shall provide operational user guidance.

Developer Note: The developer should review the assurance activities for this component to ascertain the specifics of the guidance that the evaluators will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

Content and presentation elements:

AGD_OPE.1.1C The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2C The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3C The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4C The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5C The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6C The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C The operational user guidance shall be clear and reasonable.

Evaluator action elements:

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

Assurance Activities: During operation, the activities to be described in the guidance fall into two broad categories; those that are performed by a (non-administrative) user, and those that are performed by an administrator. It should be noted that most procedures needed for non-administrative users are referenced in the assurance activities in Section 4.1. However, two additional warnings shall be provided in the guidance to users. The guidance shall warn authorized users that they must not let the storage device leave their physical control while it is powered on. Additionally, it shall state that authorized users shall not leave/store the passphrase and/or external token authorization factors with the device or if multiple factors are used, with each other.

With respect to the administrative functions, while several have also been described in section 4.1, additional information is required as follows.

The documentation must describe the process for verifying that updates to the TOE come from the intended source (which in most cases will be the TOE vendor). This verification process is initiated by the authorized user but performed by the TSF on the device. The evaluators shall verify that this process includes the following steps:

1. Instructions for obtaining the certificate that will be used by the FCS_COP.1(2) mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means and installed on the drive as part of its initial configuration. If not initially supplied on the drive, the guidance shall provide instructions on how to determine the obtained certificate can be trusted by the end user.
2. Instructions for obtaining the update itself. This should include instructions for making the update accessible (e.g., placement in a specific directory).
3. Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful.

If the TOE supports the use of external token authorization factors, the evaluator shall also check to ensure that guidance states that users are not to put any data on the external token device containing the authorization factor.

4.2.2.2 AGD_PRE.1 Preparative procedures

Developer action elements:

AGD_PRE.1.1D The developer shall provide the TOE including its preparative procedures.

Developer Note: As with the operational guidance, the developer should look to the assurance activities to determine the required content with respect to preparative procedures.

Content and presentation elements:

AGD_ PRE.1.1C The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_ PRE.1.2C The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

Evaluator action elements:

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

AGD_ PRE.1.2E The evaluator *shall apply* the preparative procedures to confirm that the TOE can be prepared securely for operation.

Assurance Activities: As indicated in the introduction above, there are significant expectations with respect to the documentation—especially when configuring the Operational environment to support TOE functional requirements. The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms (that is, combination of hardware and operating system) that the ST claims are allowed to host the TOE.

The evaluator shall check to ensure that the following guidance is provided:

- Systems (and laptops in particular) generally support a number of modes that are targeted at states of user inactivity: power management modes (e.g., hibernation, sleep/standby, auto-shutdown). There are two areas that need to be covered in the guidance.

  The first addresses the steps that must be performed to configure the platform so that the system powers down completely after a period of user inactivity; the point being that on power-down, the keying material will be erased and the hard disk is in the initial state the threat model envisions. While it is allowable for a function such as screen lock to become active due to user inactivity prior to the power-down process being initiated, it is not a substitute for power-down and does not satisfy this requirement.

  Some of the modes do not completely power down the system and shut down the operating system; instead, the system has some state stored (either in volatile memory or on disk) allowing the user to start working from where they left off prior to the mode was entered. Conformant TOEs are not allowed to enter any modes that leave the computer in a compromised state. Therefore, the power managed state must ensure that the computer is as well-protected as a full power-off state (i.e. all data is encrypted in a power managed mode, user authentication is the same as for a regular power-on). If the TOE is not claiming to provide a product that conforms to this requirement, then the TOE must include guidance that demonstrates how the TOE is not allowed to enter into any modes that do not completely power down the platform and shut down the operating system, so the second area that needs to be covered in the guidance details the steps needed to disable any modes that leave the system in a state where power is still applied to main memory of the system such that key material is present unencrypted. It shall be the case that when the hard disk cannot be accessed by a user after entering this powered-down state without first entering the required authorization factors so that the KEK is reconstructed and the DEK re-unwrapped. The guidance shall also provide instructions so that the capability to re-enable the modes is restricted to the TOE administrators.

- The TOE authorization factors cannot be used in place of the underlying Identification and Authentication mechanism for the TOE. The evaluator shall examine the guidance to ensure that if a capability exists to use the authorization factors in lieu of or as part of the platforms identification and authentication mechanism, instructions are provided on how to disable this capability, and warnings are given to the administrator that this capability is not be used in conformant TOEs.

- Instructions and information is provided to the administrator detailing how to configure the product so that all hard drives are encrypted when setting up the product, and that this is the only allowed configuration for conformant TOEs.

- As indicated in the introductory material, administration of the TOE is performed by one or more administrators that are a subset of the group of all users of the TOE. While it must be the case that the overall system (TOE plus Operational Environment) provide this capability, the responsibility for the implementation of the functionality can vary from totally the Operational Environment's responsibility to totally the TOE's responsibility. At a high level, the guidance must contain the appropriate instructions so that the Operational Environment is configured so that it provides the portion of the capability for which it is responsible. If the TOE provides no mechanism to allow separation of administrative users from the population of users, then the instructions, for instance, would cover the OS configuration of the OS I&A mechanisms to provide a unique (OS-based) identity for users, and further guidance would instruct the installer on the configuration of the DAC mechanisms of the OS using the TOE administrative identity (or identities) so that only TOE administrators would have access to the administrative executables.

  If the TOE provides some or all of this functionality, then the appropriate requirements are included in the ST from Appendix C, and the assurance activities associated with those requirements provide details on the guidance necessary for both the TOE and Operational Environment.

  The evaluators shall also perform the following tests:

  - Test 1 [Conditional]: If the separation of administrative users from all TOE users is performed exclusively through the configuration of the Operational Environment, the evaluators will, for each configuration claimed in the ST, ensure that after configuring the system according to the administrative guidance, non-administrative users are unable to access TOE administrative functions.

4.2.3 Class ATE: Tests

Testing is specified for functional aspects of the system as well as aspects that take advantage of design or implementation weaknesses. The former is done through ATE_IND family, while the latter is through the AVA_VAN family. At the assurance level specified in this PP, testing is based on advertised functionality and interfaces as constrained by the availability of design information presented in the TSS. One of the primary outputs of the evaluation process is the test report as specified in the following requirements.

4.2.3.3 ATE_IND.1 Independent testing - Conformance

Testing is performed to confirm the functionality described in the TSS as well as the administrative (including configuration and operation) documentation provided. The focus of the testing is to confirm that the requirements specified in section 4.1 are being met, although some additional testing is specified for SARs in section 4.3. The Assurance Activities identify the minimum testing activities associated with these components. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this PP.

Developer action elements:

ATE_IND.1.1D The developer shall provide the TOE for testing.

Content and presentation elements:

ATE_IND.1.1C The TOE shall be suitable for testing.

Evaluator action elements:

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

ATE_IND.1.2E The evaluator *shall test* a subset of the TSF to confirm that the TSF operates as specified.

Assurance Activities: The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the body of this PP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluators must document in the test plan that each applicable testing requirement in the ST is covered.

The Test Plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platform and the untested platforms, and make

an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. Evaluators shall especially consider OS-based mechanisms that deal with power management modes such as power-saving and hibernation functions when writing this justification. If all platforms claimed in the ST are tested, then no rationale is necessary.

The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluators are expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) is provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform.

The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include the goal of the particular procedure, the test steps used to achieve the goal, and the expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful re-run of the test, the report would show a “fail” and “pass” result (and the supporting details), and not just the “pass” result.

In performing the test activities, the evaluators will consider all of the information in the TSS to ensure that the test cases cover various operational scenarios. For instance, when the DEK is regenerated, the evaluator should make sure to examine the TSS and any AGD_PRE or AGD_OPR guidance to ensure that the requirements that all information on the disk must remain encrypted (FDP_DSK_EXT.1) and that no key material is left available (FCS_CKM_EXT.4).

4.2.4 Class AVA: Vulnerability assessment

For the first generation of this protection profile, the evaluation lab is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. Until penetration tools are created and uniformly distributed to the evaluation labs, evaluators will not be expected to test for these vulnerabilities in the TOE. The labs will be expected to comment on the likelihood of these vulnerabilities given the documentation provided by the vendor. This information will be used in the development of penetration testing tools and for the development of future protection profiles.

4.2.4.1 AVA_VAN.1 Vulnerability survey

Developer action elements:

AVA_VAN.1.1D The developer shall provide the TOE for testing.

Content and presentation elements:

AVA_VAN.1.1C The TOE shall be suitable for testing.

Evaluator action elements:

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

AVA_VAN.1.2E The evaluator *shall perform* a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3E The evaluator *shall conduct* penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

Assurance Activities: As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in disk encryption products in general, as well as those that pertain to the particular TOE . The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its non-applicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, for example, a test would be suitable at the assurance level of this PP. If exploiting the vulnerability requires an electron microscope and liquid nitrogen, for instance, then a test would not be suitable and an appropriate justification would be formulated.

4.2.5 Class ALC: Life-cycle support

At the assurance level provided for TOEs conformant to this PP, life-cycle support is limited to end-user-visible aspects of the life-cycle, rather than an examination of the TOE vendor's development and configuration management process. This is not meant to diminish the critical role that a developer's practices play in contributing to the overall trustworthiness of a product; rather, it's a reflection on the information to be made available for evaluation at this assurance level.

4.2.5.1 ALC_CMC.1 Labeling of the TOE

This component is targeted at identifying the TOE such that it can be distinguished from other products or version from the same vendor and can be easily specified when being procured by an end user.

Developer action elements:

ALC_CMC.1.1D The developer shall provide the TOE and a reference for the TOE.

Content and presentation elements:

ALC_CMC.1.1C The TOE shall be labeled with its unique reference.

Evaluator action elements:

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

Assurance Activities: The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.

4.2.5.2 ALC_CMS.1 TOE CM coverage

Given the scope of the TOE and its associated evaluation evidence requirements, this component's assurance activities are covered by the assurance activities listed for ALC_CMC.1.

Developer action elements:

ALC_CMS.2.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

ALC_CMS.2.1C The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.2.2C The configuration list shall uniquely identify the configuration items.

Evaluator action elements:

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

Assurance Activity: The "evaluation evidence required by the SARs" in this PP is limited to the information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.

5 Conformance Claims

The Conformance Claim indicates the source of the collection of requirements that is met by a PP or a Security Target (ST) that passes its evaluation. Application notes are provided in the Security Functional Requirements (SFR) and Security Assurance Requirements (SAR) sections to further clarify specific requirements that must be met.

5.1 PP Conformance Claim

This PP is conformant to CC 3.1, CC Part 2 extended and CC Part 3 conformant.

STs that claim conformance to this PP shall meet a minimum standard of strict-PP conformance as defined in Section D3 of CC Part 1 (CCMB-2006-09-001).

Strict-PP conformance means the requirements in the PP are met and that the ST is an instantiation of the PP. The ST can be broader than the PP, and in these cases the National Scheme overseeing the evaluation will approve the additions. The ST specifies that the TOE does at least the same as the PP, while the operational environment does at most the same as the PP. In this PP, assurance activities are provided to further clarify and explain the intent of the requirements specified and the expectation as to how the vendor will meet the requirements. It is expected that the evaluator of the ST will ensure strict-PP compliance by determining that the ST and its described TOE not only contain all the statements within this PP (and possibly more) but also met the expectations as stated by the assurance activities.

5.2 PP Conformance Claim rationale

This PP does not claim conformance to another PP.

6 Rationale

This section describes the rationale for the Security Objectives and Security Functional Requirements as defined within this Security Target, as well as the rationale for the assurance requirements.

6.1 Rationale for Security Functional Requirements

This section describes the rationale for the TOE Security Functional Requirements as defined in Section 4.1. In order to provide a clear expression of the degree to which the requirements implemented by the TOE mitigate the threats or implement the policies, the following text traces the requirements through the objectives to the applicable threats/policies. Table 9 illustrates the mapping from Security Functional Requirements through the Security Objectives to the Threats/Policies, with a corresponding rationale of how the requirements mitigate the threat or address the policy.

It should be noted that the following table includes references to some requirements in Appendix C; specifically those in section C.1. This is because that—unlike other requirement in Appendix C—these requirements must either be implemented by the TOE or in the operational environment in order for the TOE to meet its specified security policy. In order to aid understanding of the extent to which the threats are mitigate, these components have been put into this table. Inclusion in this table does not indicate that the TOE must implement those Appendix C, section C.1 requirements.

Table 7: Threat/Policies/Objectives/SFRs Mappings/Rationale

Threat/Policy	Objective	Rationale
T.KEYING_MATERIAL_ COMPROMISE An attacker can obtain unencrypted key material (the KEK, the DEK, authorization factors, submasks, and random numbers or any other values from which a key is derived) that the TOE has written to persistent memory and use these values to gain access to user data.	O.DEK_SECURITY The TOE will mask the DEK using a key encryption key (KEK) created from one or more submasks (which in turn are derived from the authorization factors) so that a threat agent who does not have authorization factor(s) will be unable to gain access to the user data by obtaining the DEK. (FCS_CKM.1(2), FCS_CKM.1(3), FCS_COP.1(4), FCS_RBG_EXT.1) O.KEY_MATERIAL_COMPROMISE The TOE will zeroize key material as soon as it is no longer needed to decrease the chance that such material could be used to discover a KEK or DEK. (FCS_CKM_EXT.4) O.MANAGE The TOE will 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. (FMT_SMF.1)	FCS_CKM.1 (3) levies requirements that if a passphrase authorization factor is used, that it is sufficiently conditioned to ensure an adequate KEK will be derived. FCS_CKM.1(2) is the requirement that specifies how the KEK will be derived, and specifies the key length of the KEK. This requirement allows authorization factors other than a passphrase, but mandates that the effective strength of each authorization factor is maintained. In other words, introducing another authorization factor would not weaken the strength of the conditioned passphrase. FCS_COP.1 (4) ensures a sound implementation of cryptographic operations for masking the DEK. FCS_RBG_EXT.1 ensures that keying material is robustly generated. This requirement only plays a role in satisfying this objective if the passphrase authorization factor is used or if the TOE is generating the external token authorization factor. FCS_CKM_EXT.4 plays a role in satisfying this objective by ensuring that key material is unavailable once it is no longer needed. This mitigates an attack where one may try and derive the KEK from any keying material that is discovered. FMT_SMF.1 ensures the TSF provides the functions necessary to manage

Threat/Policy	Objective	Rationale
		important aspects of the TOE. These include generating, protecting, and deleting a DEK, generating and configuring authorization factors, and configuring cryptographic functionality. The ST author may chose to incorporated other management functions if they chose.
T.PERSISTENT_INFORMATION The TOE and/or 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 The TOE will zeroize key material as soon as it is no longer needed to decrease the chance that such material could be used to discover a KEK or DEK. (FCS_CKM_EXT.4)	FCS_CKM_EXT.4 requires that all keying material is zeroized when no longer needed or upon a shutdown. This applies whether the keying material is within the cryptographic module or outside the cryptographic

Threat/Policy	Objective	Rationale
	OE.POWER_SAVE[3] The Operational environment must be configurable so that there exists at least one mechanism that will cause the system to power down after a period of time in the same fashion as the user electing to shutdown the system. Any such mechanism (e.g., sleep, hibernate) that does not conform to this requirement must be capable of being disabled by the administrator. OE.TRAINED_USERS Authorized users will be properly trained and follow all guidance for securing the TOE and authorization factors.	module (e.g., in memory). In all likelihood most keying material will be zeroized once the TSF no longer needs it, but in instances where keying material still exists, when the machine is shutting down (either manually or due to inactivity) the key material is zeroized. OE.POWER_SAVE mitigates the threat by providing a platform that can be configured so that the protection features of the TOE can be invoked. Similarly, OE.TRAINED_USERS mitigates the threat by ensuring that users shut down the system in a manner that will cause the key zeroization feature to be invoked.
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.	O.DEK_SECURITY The TOE will mask the DEK using a key encryption key (KEK) created from one or more submasks (which in turn are derived from the authorization factors) so that a threat agent who does not have authorization factor(s) will be unable to gain access to the user data by obtaining the DEK. (FCS_CKM.1(2), FCS_CKM.1(3), FCS_RBG_EXT.1, FCS_COP.1(4), FCS_COP.1(3))	FCS_CKM.1(2) and FCS_CKM.1(3) place requirements on the KEK and Authorization Factor Conditioning (respectively) to ensure they are randomized (FCS_RBG_EXT.1, FCS_COP.1(3)) to increase the difficulty in guessing the KEK/Authorization Factor) and of a length appropriate to ensure that deducing one of these values is just has difficult as deducing the DEK. The DEK is

[3] If the TOE encompasses the entire platform, this is changed to O.POWER_SAVE and "Operational Environment" is replaced with "TOE" in this column and the following column.

Threat/Policy	Objective	Rationale
		wrapped using a technique (FCS_COP.1(4)) that provides strength equivalent to the length of the DEK, thus ensuring this avenue offers no better attacks than brute force. The inability for anyone to read the key also reduces the likelihood that an attacker can obtain information that would lessen the work factor needed to deduce the key.
T.TSF_COMPROMISE A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified, or deleted).	O.CORRECT_TSF_OPERATION The TOE will provide the capability to test the TSF to ensure the correct operation of the TSF in its operational environment. (FPT_TST_EXT.1) O.TRUSTED_UPDATE The TOE shall provide administrators the capability to update the TOE firmware/software, and verify that updates to the product are received from the intended source. (FCS_COP.1(2), FCS_COP.1(3), FPT_TUD_EXT.1)	FPT_TST_EXT.1 requires self-tests to be run on the TOE (both for the cryptographic module as well as for other components) prior to the TOE being put into operation. While not providing direct protection against malicious users or processes, it provides some protection against compromise of the TSF by ensuring that the underlying mechanisms of the TSF are operating properly. An update that is verified cryptographically (FCS_COP.1(2), FCS_COP.1(3), FPT_TUD_EXT.1) mitigates the possibility that a malicious attacker will attempt to substitute a corrupted version of the TOE through the update process by ensuring that the updates are signed by a cryptographically strong

Threat/Policy	Objective	Rationale
		mechanism, and verified by the administrator prior to being installed.
T.UNAUTHORIZED_DISK_ACCESS An unauthorized user that has access to the lost hard disk may gain access to data for which they are not authorized according to the TOE security policy.	O.ENCRYPT_ALL The TOE will encrypt all data that are stored on a hard drive. (Note that this may exclude the MBR and the bootable partition that it points to.) (FDP_DSK_EXT.1 FCS_CKM.1(1) FCS_COP.1(1)) O.AUTHORIZATION The TOE must obtain the authorization factor(s) from a user to be able to decrypt the data on the hard disk. (FIA_AUT_EXT.1 FCS_CKM.1(2) FCS_COP.1(4)) O.DEK_SECURITY The TOE will mask the DEK using a key encryption key (KEK) created from one or more submasks (which in turn are derived from the authorization factors) so that a threat agent who does not have authorization factor(s) will be unable to gain access to the user data by obtaining the DEK. (FCS_CKM.1(2), FCS_CKM.1(3), FCS_COP.1(4), FCS_RBG_EXT.1,) O.KEY_MATERIAL_COMPROMISE The TOE will zeroize key material as soon as it is no longer needed to decrease the chance that such material could be used to discover a KEK or DEK. (FCS_CKM_EXT.4) O.OWNERSHIP	FDP_DSK_EXT.1 ensures the TOE performs full disk encryption, which includes all user data. "Full Disk Encryption" is defined in the Glossary for this PP as "the process of encrypting all the data on the hard drive of a computer, including the computer's OS, and permitting access to the data only after successful authentication to the FDE product" with the exception of the MBR and associated bootable partition containing the code necessary to accept and process the authentication factors. This ensures that the data are unexposed even if the device is lost. In addition to having a requirement that all the data are encrypted, FCS_CKM.1(1) and FCS_COP.1(1) specify the quality of the key used to perform the encryption, as well as the algorithm and key length that is used in the disk encryption operations. This ensures that the protection cannot be easily broken, thus ensuring the protection of the data. FIA_AUT_EXT.1 requires that users must be

Threat/Policy	Objective	Rationale
	The TOE shall ensure that ownership is taken (that is, a DEK is created, authorization factors are established, any default authorization factors are changed, a KEK is formed from the derived submasks, and the DEK is associated with the KEK) prior to any user data being accessible while the TOE is in operation. (FMT_SMF.1)	authorized by the mechanisms specified in FCS_CKM.1(2) and FCS_COP.1(4) before they are allowed access to unencrypted data from the hard drive. This ensures that unauthorized users cannot invoke the decryption mechanisms to gain access to the data. If the keys or authorization factors are compromised, then the data on the disk can be easily recovered. FCS_CKM.1(2), FCS_CKM.1(3), FCS_COP.1(4), FCS_RBG_EXT.1, and FCS_CKM_EXT.4 all ensure that the cost of obtaining key or authorization factors is as cryptographically difficult as guessing the DEK. FMT_SMF.1 addresses the threat by ensuring that there is no window in which encryption is not established on the disk drive once the TOE is put into operation. Additionally, if default authorization factors exist, there is a mechanism and appropriate guidance so that a user can change these authorization factors, thus preventing a trivial compromise of the data.
T. UNAUTHORIZED_UPDATE	O.TRUSTED_UPDATE	Updates are verified as

Threat/Policy	Objective	Rationale
A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE.	The TOE shall provide administrators the capability to update the TOE firmware/software, and verify that updates to the product are received from the intended source. (FCS_COP.1(2), FCS_COP.1(3), FPT_TUD_EXT.1)	specified in the AGD requirements and FPT_TUD_EXT.1. This verification is required to use the hash mechanism specified in FCS_COP.1(3) and the digital signature mechanisms specified in FCS_COP.1(2). Using these cryptographic mechanisms to verify the update ensure that the update comes from the intended source.
T.UNSAFE_AUTHFACTOR_VERIFICATION An attacker can take advantage of an unsafe method for performing verification of a user-entered authorization factor, resulting in exposure of the KEK, DEK, or user data.	O. SAFE_AUTHFACTOR_ VERIFICATION The TOE shall perform verification of the authorization factors in such a way that the KEK, DEK, or user data are not inadvertently exposed. (FIA_AUT_EXT.1)	FIA_AUT_EXT.1 requires the TSF to verify the authorization factors prior to the user gaining access to the data on the USB Flash Drive. It also requires that this is performed in a manner that does not provide the attacker an advantage in guessing the DEK or KEK.

6.2 Rationale for Security Assurance Requirements

The particular assurance requirements were chosen to provide an achievable level of assurance that is consistent with good commercial practices for Full Disk Encryption devices. As such, minimal additional tasks are placed upon the vendor assuming the vendor follows reasonable software engineering practices and can provide the necessary supporting guidance documents. The chosen assurance level is commensurate with the Security Problem Definition and Threats defined in Section 2. This document is intended to evolve with the ever-changing threat environment and with the advancement of development best practices and will incorporate any new requirements or assurance activities when appropriate. These advancements will be driven by actual evaluation results and vendor consortia.

Appendix A: Supporting Tables and References

[1] Common Criteria for Information Technology Security Evaluation, CCMB-2007-09, Version 3.1, September 2007.

[2] Draft Consistency Instruction Manual, for Basic Robustness Environments, Release 4.0, CC version 3.1, 2008

[3] Federal Information Processing Standard Publication (FIPS-PUB) 140-2, Security Requirements for Cryptographic Modules, National Institute of Standards and Technology, May 25, 2001 (CHANGE NOTICES (12-03-2002)

[4] Federal Information Processing Standard Publication (FIPS-PUB) 180-2, Secure Hash Standard, August 1 2002

[5] Federal Information Processing Standard Publication (FIPS-PUB) 197, Specification for the Advanced Encryption Standard (AES), November 26, 2001

[6] NIST Special Publication 800-38A, Recommendation for Block Cipher Modes of Operation, Methods and Techniques, 2001 Edition

[7] NIST Special Publication 800-38C, Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, May 2004

[8] NIST Special Pub 800-90, Recommendation for Random Number Generation Using Deterministic Random Bit Generators (Revised) , March 2007

[9] NIST Special Publication 800-108, Recommendation for Key Derivation Using Pseudorandom Functions, April 2008

[10]

[11] RFC 2898 Password-Based Cryptography Specification, Version 2.0, September 2000

[12] RFC 3394 Advanced Encryption Standard (AES) Key Wrap Algorithm, September 2002

Acronyms

AES	Advanced Encryption Standard
AF	Authorization factor
CAVS	Cryptographic Algorithm Validation System
CC	Common Criteria
CM	Configuration management
COTS	Commercial Off-The-Shelf
DEK	Data Encryption Key
DRBG	Deterministic Random Bit Generator
DoD	Department of Defense
EAL	Evaluation Assurance Level

FDE	Full Disk Encryption
FIPS	Federal Information Processing Standards
ISSE	Information System Security Engineers
IT	Information Technology
KEK	Key Encryption Key
MBR	Master Boot Record
OSP	Organization Security Policy
PP	Protection Profile
PUB	Publication
RGB	Random Bit Generator
SAR	Security Assurance Requirements
SF	Security Function
SFR	Security Functional Requirement
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	Target Security Functionality Interface
TSS	TOE Summary Specification
TOE	Target of Evaluation

Appendix B: NIST SP 800-53/CNSS 1253 Mapping

Several of the NIST SP 800-53/CNSS 1253 controls are either fully or partially addressed by compliant TOEs. This section outlines the requirements that are addressed, and can be used by certification personnel to determine what, if any, additional testing is required when the TOE is incorporated into its operational configuration.

Application Note: In this version, only a simple mapping is provided. In future versions, additional narrative will be included that will provide further information for the certification team. This additional information will include details regarding the SFR to control mapping discussing what degree of compliance is provided by the TOE (e.g., fully satisfies the control, partially satisfies the control). In addition, a comprehensive review of the specified assurance activities, and those evaluation activities that occur as part of satisfying the SARs will be summarized to provide the certification team information regarding how compliance was determined (e.g., document review, vendor assertion, degree of testing/verification). This information will indicate to the certification team what, if any, additional activities they need to perform to determine the degree of compliance to specified controls.

Since the ST will make choices as far as selections, and will be filling in assignments, a final story cannot necessarily be made until the ST is complete and evaluated. Therefore, this information should be included in the ST in addition to the PP. Additionally, there may be some necessary interpretation (e.g.,"modification") to the activities performed by the evaluator based on a specific implementation. The scheme could have the oversight personnel (e.g., Validators) fill in this type of information, or could have this done by the evaluator as part of the assurance activities. The verification activities are a critical piece of information that must be provided so the certification team can determine what, if anything, they need to do in addition to the work of the evaluation team.

Identifier	Name	Applicable SFRs
CM-5	Access Restrictions for Change	FPT_TUD_EXT.1
IA-5	Authenticator Management	FCS_CKM.1.1(Y), FIA_AUT_EXT.1, FMT_SMF.1
IA-7	Cryptographic Module Authentication	FIA_AUT_EXT.1
MP-4	Media Storage	FDP_DSK_EXT.1
SC-12	Cryptographic Key Establishment and Management	FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM_EXT.4, FMT_SMF.1
SC-13	Use of Cryptography	FCS_CKM.1.1(3), FCS_COP.1.1(1), FCS_COP.1.1(2), FCS_COP.1.1(3), FCS_COP.1.1(4), FCS_RBG_EXT.1, FMT_SMF.1
SC-28	Protection of Information at Rest	FDP_DSK_EXT.1
SI-6	Security Functionality Verification	FPT_TST_EXT.1

Appendix C: Additional Requirements

As indicated in the introduction to this PP, the baseline requirements (those that must be performed by the TOE) are contained in the body of this PP. There are additional requirements that may be included in the ST such that the TOE is still conformant to this PP; those requirements are contained in this Appendix. This Appendix has two distinct types of requirements. Those in section C.1 are required to be implemented either by the TOE or by the Operational Environment, and deal with basic cryptographic functionality that can be implemented in the Operational Environment and used by the TOE (as well as many other security products). If the TOE chooses to implement this functionality rather than relying on the Operational Environment, then these requirements will be moved to the body of the ST by the ST author.

Requirements in sections C.2 and later, on the other hand, are **not** required, but can be implemented by the TOE. In these cases, the ST author will take the appropriate information from this Appendix and include it in their ST. Note that the ST author is responsible for ensuring that requirements that may be associated with Appendix C requirements but are not listed (e.g., FMT-type requirements) are also included in the ST. Requirements not contained in this appendix are subject to review and acceptance by the National Scheme overseeing the evaluation before a conformance claim to this PP can be made.

C.1 Primary Cryptographic Requirements

As indicated in the body of this PP, it is acceptable for the TOE to either directly implement cryptographic functionality that supports the disk encryption/decryption process, or to use that functionality if it exists in the Operational Environment (for example, calling an Operating System's cryptographic provider interface; a third-party cryptographic library; or a hardware cryptographic accelerator). The requirements in this section specify the cryptographic functionality that must be present either in the TOE or the Operational Environment in order for the TOE to satisfy its security objectives. If the functionality is present in the TOE, then these requirements, as well as the assurance activities, will be moved by the ST author to the body of the ST.

If the functionality is merely used by the TOE and provided by the Operational Environment, then the developer will identify those functions in each Operational Environment listed in the ST. This identification should be such that an evaluator can use the information in the TSS (which requires that the method by which each operation is invoked is identified) coupled with the information on the functions in the Operational Environment to perform scheme-specific activities to validate that each Operational Environment listed for the TOE is able to meet the requirements in this section. The evaluator checks the Operational Environment to make sure they supply those functions and that the interfaces exist in the Operational Environment documentation. The invocation of the cryptographic services will be tested by the evaluator when performing the assurance activities described in the body of this PP.

Note: There may be additional national scheme policy imposed on the approval of the cryptography being implemented in the Operational Environment. This may include the underlying platform being evaluated or validated. The need for an additional “certification” should be verified with the sponsoring scheme.

If any of the cryptographic functionality in this section is provided by the operational environment, the ST author will include the following assumption and environmental objective in the ST.

A.STRONG_OE_CRYPTO	All cryptography implemented in the Operational Environment and used by the TOE will meet the requirements listed in Appendix C of this PP. This includes generation of external token authorization factors by a RBG.

OE.STRONG_ENVIRONMENT_ CRYPTO	The Operating Environment will provide a cryptographic function capability that is commensurate with the requirements and capabilities of the TOE and Appendix C.

C.1.1 Symmetric (Disk) Encryption

This requirement is used to specify the symmetric encryption/decryption algorithm that is used to encrypt and decrypt the data on the disk.

FCS_COP.1(1)	**Cryptographic operation (Disk Encryption)**
FCS_COP.1.1(1)	**Refinement:** The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **AES used in [selection: CBC, XTS] mode** and cryptographic key sizes **[selection: 128 bits, 256 bits]** that meet the following: **FIPS PUB 197, "Advanced Encryption Standard (AES)" and [selection: NIST SP 800-38A, NIST SP 800-38E]**.
FCS_COP.1.2(1)	DEK/IV pairs used by TOE on a platform must be unique.

Application Note: The intent of this requirement is to specify the approved AES modes that the ST author may select for AES encryption of the appropriate information on the hard disk. For the first selection, the ST author should indicate the mode or modes supported by the TOE implementation. The second selection indicates the key size to be used, which is identical to that specified for FCS_CKM.1(1). The third selection must agree with the mode or modes chosen in the first selection. If multiple modes are supported, it may be clearer in the ST if this component was iterated.

Since the CBC and XTS are block mode encryption schemes, they require IVs for encrypting the initial block of data. If the TOE uses a scheme where the disk or disks are divided into separate encryptable segments that require an IV, then all DEK/IV pairs used in encrypting data for a given platform must be unique.

Future versions of this PP may include new cryptographic modes as they are reviewed and approved by NIST.

Assurance Activities

If multiple modes are supported, the evaluator examines the TSS and guidance documentation to determine how a specific mode/key-size is chosen by the end user. The evaluator then tests each mode/key size combination in the manner found in the following sections, as appropriate. Note that some of these tests will require a reference implementation of the algorithms that is acceptable to the evaluation facility's Scheme.

The evaluator shall ensure that the TSS describes-for each supported platform listed in the ST, the process by which the disk is encrypted. If multiple disks are supported, then this description shall cover this cases as well. This description shall cover how (or if) the disk is divided into encryptable portions so that it is clear when an IV needs to be used, and the number of IVs that are potentially present in the system. The description shall also cover IV generation so that the evaluator can determine that all DEK/IV pairs used on the system are unique.

- ***CBC Mode***

The tests for CBC mode are referenced in *The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)* [AESAVS], available from http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf.

The evaluators shall run a set of known answer tests for each key size and mode supported by the TSF. Inputs are a key, IV, and either plaintext to be encrypted or ciphertext to be decrypted. All of the test vectors (both encrypt and decrypt) for these modes in the supported key lengths from http://csrc.nist.gov/groups/STM/cavp/documents/aes/KAT_AES.zip shall be used to perform these tests

The evaluators shall perform a multi-block message test for each key length and mode supported. To perform this test, the evaluators generated 10 data sets for encryption and 10 data sets for decryption. Each data set consists of key, an IV, and plaintext (for encryption) or ciphertext (for decryption). The length of a block shall be 128 bits; the length of the plaintext/ ciphertext shall be block length * i, where i indicates the data set number and i ranges from 1 to 10 (so messages will range from 128 bits to 1280 bits).

The evaluators shall perform a Monte Carlo test for each supported mode. The evaluators shall generate 10 sets of starting values for encryption (values for the key, IV, and plaintext) and 10 sets of starting values for decryption (values for the key, IV, and ciphertext). The length of the plaintext/ciphertext shall be 128 bits. Each set of starting values is used to generate and perform 100 tests; the algorithm for generating the 100 test values (per set of starting values) is contained in section 6.4.x of [AESAVS] and is dependent on the mode being tested.

- ***XTS Mode***

The XTS mode tests reference *The XTS-AES Validation System (XTSVS)* [XTSVS], available from http://csrc.nist.gov/groups/STM/cavp/documents/aes/XTSVS.pdf.

The evaluator first conducts the tests identified in the *CBC Mode* section above. After completing those tests, the evaluator examines the TSS to ensure that the range of data lengths supported in XTS mode is identified, and the format of the tweak value (either a 128-bit string or a data unit sequence number).

The evaluator then devises test sets for each key length supported. For a given key length, the evaluator chooses a sample of at least 5 data lengths to test. For each data length, the evaluator devises 100 encryption tests and 100 decryption tests. Each test is performed with a unique key, tweak, and plaintext (for encryption) or ciphertext (for decryption) value. Examples of test sets can be seen in http://csrc.nist.gov/groups/STM/cavp/documents/aes/XTSTestVectors.zip.

C.1.2 Signature Verification

This requirement is used to perform verification of the signature on the updates to the TOE.

FCS_COP.1(2)	**Cryptographic operation (Signature Verification)**
FCS_COP.1.1(2)	**Refinement:** The TSF shall perform **cryptographic signature verification for TOE updates** in accordance with a ***[selection:*** ***(1) Digital Signature Algorithm (DSA) with a key size (modulus) of 2048 bits or greater,*** ***(2) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater, or*** ***(3) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater]*** that meets the following: **Case: Digital Signature Algorithm** • **[FIPS PUB 186-3, "Digital Signature Standard"** **Case: RSA Digital Signature Algorithm** • **FIPS PUB 186-3, "Digital Signature Standard"** **Case: Elliptic Curve Digital Signature Algorithm** • **FIPS PUB 186-3, "Digital Signature Standard"**

- **The TSF shall implement "NIST curves" P-256, P-384 and [selection: P-521, no other curves] (as defined in FIPS PUB 186-3, "Digital Signature Standard").**

Application Note: The ST Author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement should be iterated to specify the functionality. For the algorithm chosen, the ST author should make the appropriate assignments/selections to specify the parameters that are implemented for that algorithm.

For elliptic curve-based schemes, the key size refers to the $\log_2$ of the order of the base point. As the preferred approach for digital signatures, elliptic curves will be required in future publications of this PP.

Assurance Activities: This requirement is used to verify digital signatures attached to updates from the TOE manufacturer before installing those updates on the TOE. Because this component is to be used in the update function, additional assurance activities to those listed below are covered in other assurance activities section in this document. The following assurance requirements deal only with the implementation for the digital signature algorithm; the evaluator performs the testing appropriate for the algorithm(s) selected in the component.

Hash functions and/or random number generation required by these algorithms must be specified in the ST; therefore the assurance activities associated with those functions is contained in the associated Cryptographic Hashing and Random Bit Generation sections. Additionally, the only function required by the TOE is the verification of digital signatures. If the TOE generates digital signatures to support the implementation of any functionality required by this PP, then the cognizant evaluation and validation scheme must be consulted to determine the required assurance activities.

For any algorithm, the evaluators check the TSS to ensure that it describes the overall flow of the signature verification. This should at least include identification of the format and general location (e.g., "firmware on the hard drive device" vice "memory location 0x00007A4B") of the data to be used in verifying the digital signature; how the data received from the operational environment are brought on to the device; and any processing that is performed that is not part of the digital signature algorithm (for instance, checking of certificate revocation lists).

Each section below contains the tests the evaluators must perform for each type of digital signature scheme. Based on the assignments and selections in the requirement, the evaluators choose the specific activities that correspond to those selections.

It should be noted that for the schemes given below, there are no key generation/domain parameter generation testing requirements. This is because it is not anticipated that this functionality would be needed in the end device, since the functionality is limited to checking digital signatures in delivered updates. This means that the domain parameters should have already been generated and encapsulated in the hard drive firmware or on-board non-volatile storage. If key generation/domain parameter generation is required, the evaluation and validation scheme must be consulted to ensure the correct specification of the required assurance activities and any additional components.

Similarly, it is not anticipated that signature generation will be required to meet the baseline requirements of this PP. If signature generation is required, then the evaluation and validation scheme must be consulted to ensure the correct specification of the required assurance activities and any additional components.

- ***RSA***

There are two options for implementing the signature generation/verification function: ANSI X9.31 and PKCS #1 (Version 1.5 and/or Version PSS). At least one of these options must be implemented. Each implemented version must be tested as indicated below. If PKCS #1 Version PSS is chosen, then the evaluator shall check the TSS to ensure the length of the salt is specified.

If the TOE supports more than one modulus size, then the evaluator shall perform the following test for all modulus sizes. If the TOE supports more than one hash algorithm, the evaluator shall perform the following test for all hash algorithms. This means that if the implementation allows the choice of 2 modulus sizes and 2 hash algorithms, the evaluator would perform the following test 4 times.

The evaluator shall generate three groups of data. Each group of data consists of a modulus and 4 sets of test vectors consist with the modulus. The test vectors consist of a public exponent *e*; a psuedorandomly-generated message; and a signature for the message using the associated private key (consistent with *e* and the modulus *n*). This means that there will be a minimum of 12 test vectors for each modulus size/hash algorithm supported by the TSF.

In 3/4s of the test vectors after the correct signature has been generated (but not "fed" to the TSF), the evaluators will alter the public key, message, or signature (making sure to do at least 2 of each) so that the signature verification failure function will be tested. The evaluators shall then run the test vectors through the TSF and verify that the results are correct.

In addition, if the algorithm implemented is RSASSA-PKCS1-v1_5, as specified in *Public Key Cryptography Standards (PKCS) #1 v2.1: RSA Cryptography Standard-2002*, or the RSA algorithm described in X9.31, *Digital Signatures Using*

Reversible Public Key Cryptography for the Financial Services Industry (rDSA), the appropriate additional test vectors from http://csrc.nist.gov/groups/STM/cavp/documents/dss/SigVer15EMTest.zip (for PKCS #1 Version 1.5 implementations) or http://csrc.nist.gov/groups/STM/cavp/documents/dss/SigVer931IRTest.zip (for X9.31 implementations) shall be used by the evaluators to verify that the implementation successfully passes these tests.

- ***DSA***

The evaluator examines the TSS to ensure that the values used for (L, N) are given, and the hash algorithm(s) used are specified. The evaluator verifies that the hash algorithm used for a specific (L,N) provides the requisite strength, as specified in Tables 2 and 3, Section 5.6.1 of SP 800-57, *Recommendation for Key Management --Part I: General (Revised)*. The evaluators shall also ensure that the (L,N) selected has comparable strength to the symmetric (data) encryption algorithm used on the USB Flash Drive (e.g., if 128-bit AES is used to encrypt the user data, then an (L,N) of at least (3072, 256) is required).

The evaluator performs the following test for each (L,N) and hash combination supported. The evaluator shall generate a key pair. The evaluators will then pseudorandomly generate 15 1024-bit messages and sign them with the private key. For about half of the messages--after the correct signature has been generated (but not "fed" to the TSF)--the evaluators will alter the public key, message, or signature (making sure to do at least 2 of each) so that the signature verification failure function will be tested. The evaluators shall then run the test vectors through the TSF and verify that the results are correct.

- ***ECDSA***

The evaluators shall examine the TSS to determine the curve or curves used in the implementation are specified and consistent with the requirement, and the hash or hashes supported are specified. The evaluators shall conduct the following test for each curve, hash pair implemented by the TSF.
The evaluator generates 15 sets of data. Each dataset consists of a pseudorandom message; a public/private key pair (d,Q), and a signature (r,s). For about half of the messages--after the correct signature has been generated (but not "fed" to the TSF)--the evaluators will alter the public key, message, or signature (making sure to do at least 2 of each) so that the signature verification failure function will be tested. The evaluators shall then run the data through the TSF and verify that the results are correct.

C.1.3 Cryptographic Hashing

This requirement specifies the hashing used to perform a variety of higher-level activities, such as signature verification, passphrase conditioning, etc.

FCS_COP.1(3)	**Cryptographic operation (Cryptographic Hashing)**
FCS_COP.1.1(3)	**Refinement:** The TSF shall perform **cryptographic hashing services** in accordance with **[selection: SHA-1, SHA 224, SHA 256, SHA 384, SHA 512]** and **message digest sizes [selection: 160, 224, 256, 384, and 512] bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard"*.
Application Note:	The intent of this requirement is to specify the hashing function. The hash selection must support the message digest size selection. The hash selection should be consistent with the overall strength of the algorithm used for FCS_COP.1(1) and FCS_COP.1(2) (SHA 256 for 128-bit keys, SHA 512 for 256-bit keys). In subsequent publications of this PP, it is likely that SHA-1 will no longer be an approved algorithm for cryptographic hashing.
Assurance Activities:	The evaluator checks the AGD documents to determine that any configuration that is required to be done to configure the functionality for the required hash sizes is present. The evaluator shall check that the association of the hash function with other TSF cryptographic functions (for example, the digital signature verification function) is documented in the TSS. The cryptographic hashing tests reference *The Secure Hash Algorithm Validation System (SHAVS)* [SHAVS], available from http://csrc.nist.gov/groups/STM/cavp/documents/shs/SHAVS.pdf. The TSF hashing functions can be implemented in one of two modes. The first mode is the byte-oriented mode. In this mode the TSF only hashes messages that are an integral number of bytes in length; i.e., the length (in bits) of the message to be hashed is divisible by 8. The second mode is the bit-oriented mode. In this mode the TSF hashes messages of arbitrary length. As there are different tests for each mode, an indication is given in the following sections for the bit-oriented vs. the byte-oriented tests. The evaluator shall perform all of the following tests for each hash algorithm implemented by the TSF and used to satisfy the requirements of this PP. • **Short Messages Test - Bit-oriented Mode** The evaluators devise an input set consisting of *m*+1 messages, where *m* is the block length of the hash algorithm. The length of the messages range sequentially from 0 to *m* bits. The message text shall be psuedorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

- **Short Messages Test - Byte-oriented Mode**

The evaluators devise an input set consisting of *m/8*+1 messages, where *m* is the block length of the hash algorithm. The length of the messages range sequentially from 0 to *m*/8 bytes, with each message being an integral number of bytes. The message text shall be psuedorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

- **Selected Long Messages Test - Bit-oriented Mode**

The evaluators devise an input set consisting of *m* messages, where *m* is the block length of the hash algorithm. The length of the *i*th message is $512 + 99*i$, where $1 \leq i \leq m$. The message text shall be pseudorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

- **Selected Long Messages Test - Byte-oriented Mode**

The evaluators devise an input set consisting of *m/8* messages, where *m* is the block length of the hash algorithm. The length of the *i*th message is $512 + 8*99*i$, where $1 \leq i \leq m/8$. The message text shall be psuedorandomly generated. The evaluators compute the message digest for each of the messages and ensure that the correct result is produced when the messages are provided to the TSF.

- **Pseudorandomly Generated Messages Test**

This test is for byte-oriented implementations only. The evaluators randomly generate a seed that is *n* bits long, where *n* is the length of the message digest produced by the hash function to be tested. The evaluators then formulate a set of 100 messages and associated digests by following the algorithm provided in Figure 1 of [SHAVS]. The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

C.1.4 Key Masking

This requirement specifies the operations to be used to mask the DEK using the KEK. If intermediate keys are used, the ST author iterates this requirement to specify the masking used in those cases.

FCS_COP.1(4)	**Cryptographic operation (Key Masking)**
FCS_COP.1.1(4)	**Refinement:** The TSF shall perform **key masking** in accordance with a specified cryptographic algorithm **[selection: XOR; AES used in ECB mode; AES used in CCM mode; AES Key Wrap; AES Key Wrap with Padding]** and the cryptographic key size **[selection: 128 bits, 256 bits]** that meet the following: [**selection: "None" for XOR; FIPS PUB 197, *Advanced Encryption Standard (AES)* for AES and NIST SP 800-38A for ECB mode and NIST SP 800-38C for CCM**

mode; NIST SP 800-38F for Key Wrap].

Application Note: In the first selection, the ST author chooses the method by which the KEK is used to mask the DEK: either XORing the KEK and the DEK, using AES in ECB mode, using AES in CCM mode, or using one of the two AES-based Key Wrap methods specified in NIST SP 800-38F. Based on this selection, the last selection should be used to select the appropriate reference ("none" is used of XOR). The second selection should be made to reflect the size of the KEK.

NIST SP 800-38F is currently in pre-publication; once formally published, the published version applies to new evaluations using this PP.

Assurance Activities: If the DEK masking method is using “XOR”, the evaluators shall verify that the use of XOR is stated in the TSS.

If AES in ECB mode is used, then the following assurance activities will be performed.

The evaluator shall ensure that the vendor has described the method/algorithm by which the KEK is used to mask the DEK using AES (for example, any options specified by the FIPS documents are identified, methods of padding the input, truncating the output, etc.).

The evaluator shall perform the following tests. If multiple modes are supported, the evaluator examines the TSS and guidance documentation to determine how ECB and the specified key-size is chosen by the end user. The evaluator then tests each key size in the manner found in the following sections, as appropriate. Note that some of these tests will require a reference implementation of the algorithms that is acceptable to the evaluation facility's Scheme.

The ECB mode tests reference *The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)* [AESAVS], available from http://csrc.nist.gov/groups/STM/cavp/documents/aes/AESAVS.pdf.

The evaluators shall run a set of known answer tests for each key size supported by the TSF. Inputs are a key and either plaintext to be encrypted or ciphertext to be decrypted. All of the test vectors (both encrypt and decrypt) for ECB mode in the supported key lengths from http://csrc.nist.gov/groups/STM/cavp/documents/aes/KAT_AES.zip shall be used to perform these tests

The evaluators shall perform a multi-block message test for each key length supported. To perform this test, the evaluators generated 10 data sets for encryption and 10 data sets for decryption. Each data set consists of key and plaintext (for encryption) or ciphertext (for decryption). The length of a block shall be 128 bits; the length of the plaintext/ ciphertext shall be block length * i, where i indicates the data set number and i ranges from 1 to 10 (so messages will range from 128 bits to 1280 bits).

The evaluators shall perform a Monte Carlo test. The evaluators shall generate 10 sets of starting values for encryption (values for the key and plaintext) and 10 sets of starting values for decryption (values for the key and ciphertext). The length of the plaintext/ciphertext shall be 128 bits. Each set of starting values is used to generate and perform 100 tests; the algorithm for generating the 100 test values (per set of starting values) is contained in section 6.4.1 of [AESAVS]. If AES in CCM mode is used, then the following assurance activities will be performed.

The evaluator shall ensure that the vendor has described the method/algorithm by which the KEK is used to mask the DEK using AES (for example, any options specified by the FIPS documents are identified, methods of padding the input, truncating the output, etc.).

The evaluator shall test the generation-encryption and decryption-verification functionality of AES-CCM for the following input parameter and tag lengths:

128 bit and 256 bit keys

Two payload lengths. One payload length shall be the shortest supported payload length, greater than or equal to zero bytes. The other payload length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits).

Two or three associated data lengths. One associated data length shall be 0, if supported. One associated data length shall be the shortest supported payload length, greater than or equal to zero bytes. One associated data length shall be the longest supported payload length, less than or equal to 32 bytes (256 bits). If the implementation supports an associated data length of 216 bytes, an associated data length of 216 bytes shall be tested.

Nonce lengths. All supported nonce lengths between 7 and 13 bytes, inclusive, shall be tested.

Tag lengths. All supported tag lengths of 4, 6, 8, 10, 12, 14 and 16 bytes shall be tested.

To test the generation-encryption functionality of AES-CCM, the evaluator shall perform the following four tests:

Test 1. For EACH supported key and associated data length and ANY supported payload, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

Test 2. For EACH supported key and payload length and ANY supported associated data, nonce and tag length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

Test 3. For EACH supported key and nonce length and ANY supported associated data, payload and tag length, the evaluator shall supply one key value and 10 associated data, payload and nonce value 3-tuples and obtain the resulting ciphertext.

Test 4. For EACH supported key and tag length and ANY supported associated data, payload and nonce length, the evaluator shall supply one key value, one nonce value and 10 pairs of associated data and payload values and obtain the resulting ciphertext.

To determine correctness in each of the above tests, the evaluator shall compare the ciphertext with the result of generation-encryption of the same inputs with a known good implementation.

To test the decryption-verification functionality of AES-CCM, for EACH combination of supported associated data length, payload length, nonce length and tag length, the evaluator shall supply a key value and 15 nonce, associated data and ciphertext 3-tuples and obtain either a FAIL result or a PASS result with the decrypted payload. The evaluator shall supply 10 tuples that should FAIL and 5 that should PASS per set of 15.

If AES Key Wrap or AES Key Wrap with padding is used, then the following assurance activities shall be performed.

The evaluation team shall check the TSS to ensure it describes how the key wrapping and unwrapping is performed. This description can reference sections of 800-38F to point out how the cryptographic operations are performed.

C.1.5 Random Bit Generation

This requirement specifies the requirements for the Random Bit Generation function required in several cryptographic operations, including the creation of the DEK.

FCS_RBG_EXT.1 Extended: Cryptographic operation (Random Bit Generation)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST Special Publication 800-90 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES), Dual_EC_DRBG (any)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] 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_RBG_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 security strength of the keys and hashes that it will generate.

Application Note: NIST Special Pub 800-90B describes the minimum entropy measurement that will probably be required future versions of FIPS-140. If possible this should be used immediately and will be required in future publications of this PP.

For the first selection in FCS_RBG_EXT.1.1, the ST author should select the standard to which the RBG services comply (either 800-90B or 140-2 Annex C).

SP 800-90B contains four different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used (if 800-90B is selected), and include the specific underlying cryptographic primitives used in the requirement or in the TSS. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CT_DRBG are allowed. While any of the curves defined in 800-90B are allowed for Dual_EC_DRBG, the ST author not only must include the curve chosen, but also the hash algorithm used.

For the second selection in FCS_RBG_EXT.1.1, the ST author indicates whether the sources of entropy are software-based, hardware-based, or both. If there are multiple sources of entropy, the ST will elaborate each entropy sources and whether it is hardware- or software-based. Hardware-based noise sources are preferred.

Note that for FIPS Pub 140-2 Annex C, currently only the method described in *NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms*, Section 3 is valid. If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS_COP.1 may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG.

The ST author also ensures that any underlying functions are included in the baseline requirements for the TOE.

For the selection in FCS_RBG_EXT.1.2, the ST author selects the appropriate number of bits of entropy that corresponds to the greatest security strength of the algorithms included in the ST. Security strength is defined in Tables 2 and 3 of NIST SP 800-57A. For example, if the implementation includes 2048-bit RSA (security strength of 112 bits), AES 128 (security strength 128 bits), and HMAC-512 (security strength 256 bits), then the ST author would select 256 bits.

Assurance Activity: Documentation shall be produced—and the evaluator shall perform the activities—in accordance with Annex G, Entropy Documentation and Assessment.

The evaluator shall perform the following tests, depending on the standard to which the RBG conforms.

- **Implementations Conforming to FIPS 140-2, Annex C**

The reference for the tests contained in this section is *The Random Number Generator Validation System (RNGVS)* [RNGVS]. The evaluators shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme.

The evaluators shall perform a Variable Seed Test. The evaluators shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluators ensure that the values returned by the TSF match the expected values.

The evaluators shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluators shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluators then invoke the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in *NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms*, Section 3. The evaluators ensure that the 10,000th value produced matches the expected value.

- **Implementations Conforming to NIST Special Publication 800-90**

The evaluator shall perform 15 trials for the RNG implementation. If the RNG is configurable, the evaluator shall perform 15 trials for each configuration. The

evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RNG functionality.

If the RNG has prediction resistance enabled, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90).

If the RNG does not have prediction resistance, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) reseed, (4) generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call.

The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator.

Entropy input: the length of the entropy input value must equal the seed length.

Nonce: If a nonce is supported (CTR_DRBG with no df does not use a nonce), the nonce bit length is one-half the seed length.

Personalization string: The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied.

Additional input: the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.

C.1.6 Random Bit Generation

This requirement specifies the requirements for the Random Bit Generation function required in several cryptographic operations, including the creation of the DEK.

C.2 TOE Identification and Authentication

In the body of the PP, the TOE is not required to perform Identification and Authentication (I&A). While it is responsible for being able to accept and process authorization factors, this is not treated as "traditional" I&A. The TOE is required to provide management functions, but it is acceptable for the TOE to depend on the Operational environment to control access to the TOE management capability and this is what is currently specified in the PP.

However, if the TOE does provide some level of I&A functionality, then that should be specified by the ST Author through the use of the following information. Information describing the capability provided by the TOE is pulled from the following, ensuring that Security Problem Description information, Objectives, Rationale, Requirements (and associated Assurance Activities), and 800-53/CISSP 1253 information are included in the ST.

If the TOE does maintain the notion of "administrator" and enforces access to the management function based on this maintained identity, but does not provide its own mechanisms to establish this identity (for example, if it depends on information passed by the Operating System to establish the identity of the user), then a subset of the requirements below will need to be included. In this case, the ST author must include the appropriate subset of the information below and make appropriate adjustments in the body of the PP. These will be reviewed by the National Scheme overseeing the evaluation to determine that they result in an ST that will be compliant with this PP.

There is no requirement that the TOE provide any granularity with respect to users that perform I&A to the TOE and the management capabilities that they are able to perform (e.g., an administrator that is restricted to creating passphrases for passphrase-based authentication factors vs. an administrator that can also enable and disable encryption). If such a capability is implemented and a vendor wants to claim that capability in the ST, then appropriate additions to the Threats, Objectives, Rationale, SFRs, and Assurance Activities need to be drafted and proposed to the PP maintenance organization.

Table C.1-1: Additions to Organizational Security Policies

Policy	Policy Description	Formal Organizational Policy Reference
P.I_AND_A	All users must be identified and authenticated before accessing any controlled resources with the exception of public objects.	DODI 8500.2 Enclosure 4, Attachment 4 IAIA-2

Table C.1-2: Additions to Security Objectives for the TOE

Objective	Objective Description
O.IDAUTH	The TOE will identify and authenticate authorized administrators before allowing them access to the TOE management functions.

In addition to adding O.IDAUTH, the ST author must also change the environmental objective OE.RESTRICTED_FUNCTIONS to an objective for the TOE: O.RESTRICTED_FUNCTIONS and place it in the appropriate table in the body of the ST.

Table C.1.-3: Additions to Security Objectives to Threats and Policies Mappings

Threat/Policy	Objectives Addressing the Threat and Policies	Rationale
P.I_AND_A All users must be identified and authenticated before accessing any controlled resources with the exception of public objects.	O.IDAUTH The TOE will identify and authenticate authorized administrators before allowing them access to the TOE management functions.	A.PLATFORM_I&A requires that the underlying OS performs identification and authentication after the user has correctly entered the required authorization factors. However, only a subset of users possessing authorization factors are allowed to perform TOE management functions. O.IDAUTH implements this restriction by requiring an authorized TOE administrator to perform I&A to the TOE prior to invoking any of the management functions.

In addition to adding the above rationale, the ST author must also change the environmental objective OE.RESTRICTED_FUNCTIONS to an objective for the TOE: O.RESTRICTED_FUNCTIONS in the Security Objectives to Threats mapping (the text can be left as is, since the only item changing is the entity that is responsible for implementing the objective).

Table C.1-4: Additions to Rationale for TOE Security Functional Requirements

Objective	Requirement Addressing the Objective	Rationale
O.RESTRICTED_FUNCTIONS Management functions will be limited to an authorized administrator.	FIA_UID.2 FIA_UAU.2 FMT_MOF.1	TOE Users are defined as those individuals that possess valid authorization factors; this allows decryption of the information on the disk. This set of users can also perform I&A to the underlying OS. In terms of the TOE, though, only authorized administrators will possess valid I&A

		credentials to access the TOE. Since every user that is able to successfully log in to the TOE is an administrator, the requirement that no management functions can be performed prior to I&A being successfully completed is sufficient to implement the objective. It should be noted that "TSF-mediated actions on behalf of that administrator" refers to the management functions defined in FMT_SMF, and not to functions that take place prior to the boot (formation of the KEK, for instance) or the on-the-fly cryptographic operations to and from the hard disk.

User Identification (FIA_UID)

FIA_UID.2	**User identification before any action**
FIA_UID.2.1	**Refinement:** The TSF shall require each **authorized administrator** to be successfully identified before allowing any other TSF-mediated actions on behalf of that **administrator**.
Application Note:	Note that once disk encryption is initialized, an administrator has to possess valid authorization factors to administer the TOE since the DEK is required to be encrypted by a KEK, which is in turn required to be encrypted by a key derived from the authorization factors.
Assurance Activities:	Because the identification and authentication are both performed by the TOE in a sequential fashion, the assurance activities for both this component and FIA_UAU.2 are discussed in the FIA_UAU.2 assurance activities section.

User Authentication (FIA_UAU)

FIA_UAU.2	**User authentication before any action**
FIA_UAU.2.1	**Refinement:** The TSF shall require each **authorized administrator** to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that **administrator**.
Assurance Activities:	As indicated for FIA_UID.2, the assurance activities here cover both the FIA_UID.2 and FIA_UAU.2 components.

The evaluator shall review the AGD guidance for a discussion of how the administrators are established for the TOE. There will be instructions for creating user identifiers as well as initial authentication information for the users. The evaluators shall determine that the guidance also includes instructions for invoking the I&A mechanisms, as well as (if the capability is provided) instructions for changing the authentication information (e.g., a user changing their passphrase). The configuration guidance will also contain information on any platform configuration needed to protect the information used by the TOE to perform the I&A functions.

The evaluator shall review the TSS section to determine that it is consistent in its description of the I&A mechanisms in terms of their capabilities and use. The TSS section will also detail how the actual management capabilities are protected from invocation prior to the I&A process being successfully completed. As part of this analysis, the evaluator shall use the information in the TSS section identifying non-TOE products used in administration and ensure that a discussion exists detailing how these products cannot successfully be invoked directly in order to manage the TOE without first performing I&A to the TOE.

The evaluator shall also perform the following tests:

- Test 1: Ensure that administrator IDs can be established and used to successfully log in to the TOE to invoke the TOE management functions.

- Test 2: Ensure that incorrectly entering the user ID and/or authentication information results in the inability of the user to invoke the TOE management functions.

- Test 3: Ensure that the TOE management functions cannot be directly invoked, bypassing the I&A process.

C.3 Supporting Cryptographic Requirements

Several selections in the key management cryptographic requirements that reference standards will require that the TOE implement additional cryptographic functionality over and above what is specified in the main body of the PP. As the ST is being created, if the ST author chooses a selection that references such a standard, this section will contain the additional SFRs and associated assurance activities that will be needed in the main body of the ST.

If these requirements are included in the ST, then the ST author will determine the existing FCS_COP requirements that these functions support, and update the appropriate objective to requirements rationale section appropriately (in most cases this will be O.AUTHORIZATION). No updates to the objectives are needed since these requirements are support requirements.

C.3.1 HMAC Function

The HMAC function is used for implementing the NIST SP 800-90 HMAC_DRBG function as well as the PRF in NIST SP 800-132. Note that it also requires the use of a SHA function, so if this requirement is used in the ST, then the hashing requirement in C.1.2 must be included as well, with the appropriate selections. It should be noted that since the RBG function is required to be implemented on the USB flash drive, the mechanism satisfying this requirement must be implemented on the USB flash drive. It is expected that just one key-length/hash function/block size/output MAC length is used. If any of these parameters can be configured, then this requirement should be iterated in the ST to reflect this.

FCS_COP.1	**Cryptographic operation (Keyed Cryptographic Hashing)**
FCS_COP.1.1	**Refinement:** The TSF shall perform **keyed cryptographic hashing services** in accordance with **[The Keyed-Hash Message Authentication Code]** and cryptographic key size **[selection: 128 bits, 256 bits]** that meet the following: **FIPS 198-1.**
Application Note:	The selection in this requirement must be consistent with the key size specified for the size of the DEK.
Assurance Activities:	The evaluator shall examine the TSS to ensure that it specifies the following values used by the HMAC function: key-length, hash function used, block size, and output MAC length used. The evaluator shall also perform a Random Message Test, referenced in The Keyed-Hash Message Authentication Code Validation System (HMACVS) [HMACVS] available from http://csrc.nist.gov/groups/STM/cavp/documents/mac/HMACVS.pdf. For the test, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall ensure that the HMAC produced by the TSF agrees with the expected value.

C.4 Authorization Factors

The requirements in this section are used to specify appropriate functionality and additional functionality depending on the selections in FCS_CKM.1.1(2). If passphrases are selected, this section contains the requirement for passphrase composition and conditioning that the TOE must meet (FCS_CKM.1(Y)). If a TPM is used to provide an authorization factor, then the following assumption must be included in the ST, as well as the Operational Environment Objective and mapping.

Assumption	Description of Assumption
A.TPM_PIN_ANTI-HAMMER	Authorization factors stored on a TPM device must be protected by a PIN, and the TPM device must implement an anti-hammer capability to prevent brute-force guessing attacks.

Assumption	Objectives Addressing the Assumption	Rationale
A.TPM_PIN_ANTI-HAMMER Authorization factors stored on a TPM device must be protected by a PIN, and the TPM device must implement an anti-hammer capability to prevent brute-force guessing attacks.	OE.PROTECT_TPM_AF The Operating Environment will ensure that any TPM-based authorization factors are protected by a PIN that is of appropriate strength for the using organization, and that this TPM implements anti-hammer countermeasures to mitigate brute force attacks on the PIN.	Since PINs implemented by TPMs are significantly shorter than the 128-bit or 256-bit authorization factor they are protecting, an attacker to brute force guess the PIN using much less work than brute force guessing the authorization factor it is protecting. The Operational Environment must ensure that anti-hammer measures are in place, therefore, to mitigate these types of attacks.

The TOE is not required to generate either external authorization factors or passphrase-based authorization factors. However, if a TOE does offer this service then components in this section will need to be included in the ST for the TOE to claim credit for this capability, and the appropriate selection made from item "c" in FMT_SMF.1.1 to reflect that the administrator is the one that generates such authorization factors.

FCS_CKM.1(Y) **Cryptographic key generation (Passphrase formation and conditioning)**

FCS_CKM.1.1(Y) **Refinement: The TSF shall accept passphrases that are used to generate a submask that contain at least [assignment: positive integer of 64 or more] characters in the set of {upper case characters, lower case characters**, **and** [assignment: *other supported characters*]} **and shall be conditioned [selection:**

- **using [selection: SHA-1, SHA-256, SHA-512] for 128-bit DEKs;**
- **using [selection: SHA-256, SHA-512] for 256-bit DEKs;**
- **using NIST SP 800-132 with a salt generated using a Random Bit Generator as specified in FCS_RBG_EXT.1, an iteration count of [assignment: *number greater than or equal to 1000*], and HMAC using [selection: SHA-1, SHA-256, SHA-512];**

] such that the output of the conditioning function is at least equal to the size (in number of bits) of the DEK.

Application Note: A passphrase is a sequence of words taken from a dictionary of words in a random fashion such that they provide necessary entropy to generate the submask derived from the passphrase. Ideally this would be performed by the

TOE, but for this version of the PP there is no requirement that the TOE be able to compose passwords in this fashion. Rather, this component places requirements on the passphrases the TOE must support in terms their minimum length. The ST author fills in the assignment with the number of characters that are supported for passphrases; this has to be at least 64 (so at a minimum a 64-character passphrase must be able to be entered and processed by the TOE). The string that results consists of a sequence of characters encoded in a scheme determined by the underlying OS. This sequence must be conditioned into a string of bits that forms the submask to be used as input into the KEK. Conditioning can be performed using one of the identified hash functions or the process described in NIST SP 800-132; the method used is selected by the ST Author. If 800-132 conditioning is specified, then the ST author will fill in the number of iterations (C) that are performed; this value must be at least 1000. 800-132 also requires the use of a pseudo-random function (PRF) consisting of HMAC with an approved hash function. The ST author selects the hash function used, also includes the appropriate requirements for HMAC and the hash function from Appendix C.

In subsequent publications of this PP, it is likely that SHA-1 will no longer be an approved algorithm for cryptographic hashing, and that conditioning using SP 800-132 will be required.

Assurance Activity: There are two aspects of this component that require evaluation: the minimum length of a passphrase as specified in the selection is supported, and the characters that are input are subject to the selected conditioning function. These activities are separately addressed in the text below.

Support for minimum passphrase length

The evaluators shall check the TSS section to determine that it specifies that a capability exists to accept passphrases with the smaller of 64 or the number in the assignment characters. The evaluators shall also check the operational guidance to determine that there are instructions for administrators generating ~~such~~ passphrases, and that guidance indicates how the passphrases are entered into the TOE.

In addition to the analysis above, the evaluator shall also perform the following tests on a TOE configured according to the AGD_PRE guidance:

- Test 1: The evaluator shall compose several passphrases that comply with the operational guidance. The evaluators shall ensure at least one of these passphrases is the maximal length as specified in the assignment statement in the requirement. For each composed passphrase, the evaluator shall demonstrate that it is accepted by the TOE and can be used as an authorization factor (that is, the evaluator is able to unlock the disk using this passphrase).
- Test 2 [conditional]: If the ST author has filled in additional supporting characters in the 2nd assignment, ensure that the TOE contains support

for passphrases composed as specified in guidance contained in the AGD_OPR or AGD_PRE guidance with respect to the specified special characters. For instance, if the guidance specifies that passphrases must contain a special character, this test would fail if the TOE only supported letters and numbers. The evaluator ensures that all characters indicated in the requirement can be used in a valid passphrase.

Passphrase Conditioning

For SHA-based conditioning of the passphrase, the evaluator performs the following activities. The evaluator shall check that the TSS describes the method by which the passphrase is first encoded and then fed to the SHA algorithm. The settings for the algorithm (padding, blocking, etc.) shall be described, and the evaluator shall verify that these are supported by the selections in this component as well as the selections in FCS_COP.1(3) concerning the hash function itself. The evaluator shall verify that the TSS contains a description of how the output of the hash function is used to form the submask that will be input into the function described in FCS_CKM.1(2), and is at least the same length as the DEK as specified in FCS_CKM.1(1).

For 800-132-based conditioning of the passphrase, the required assurance activities will be performed when doing the assurance activities for the appropriate Appendix C requirements. If any manipulation of the master key is performed in forming the submask that will be used to form the KEK, that process shall be described in the TSS.

No explicit testing of the formation of the submask from the input passphrase is required.

FCS_CKM_EXT.1(X) Cryptographic key generation (External token authorization factor generation)

FCS_CKM_EXT.1.1(X) The TSF shall derive an external authorization factor generated by a Random Bit Generator as specified in FCS_RBG_EXT.1 that produces an authorization factor of [selection: 128 bits, 256 bits] that was seeded with entropy at least equal to the size of the DEK, as specified in FCS_CKM.1(1).

FCS_CKM_EXT.1.2(X) The TSF shall be able to store the authorization factor on [selection: an external device, a TPM].

Application Note: The first selection should indicate an identical number of bits as specified for the DEK in FCS_CKM.1(1). The second selection should be used to specify the type of device that the TOE shall be capable of storing the generated authorization factor on. If the TOE does only the generation of the authorization factor and depends on the Operational Environment to store the authorization factor on the appropriate device, then it is acceptable to create a

new component that contains just the first element of this component and include it in the ST. All assurance activities will apply to this case.

Assurance Activity: The evaluator reviews the guidance documentation to confirm that the steps necessary for an administrator to generate an external authorization factor are described. The evaluator reviews the TSS portion of the ST to confirm that the external authorization factor generation process is described, including how the generation function uses the RBG, and how the RBG function is seeded. Finally, the evaluator reviews the TSS section (or administrative guidance documentation) to determine how the value generated by the RBG is transferred to the device specified in the selection. It should be noted that the RBG used must be provided by the TOE and meet the requirements specified in FCS_RBG_EXT.1 in order for this requirement to be claimed in the ST.

The following tests must be performed by the evaluator:

- Test 1: Following the administrative guidance, create an external token authorization factor. If possible, confirm the # of bits the authorization factor contains. Load the external authorization factor into its "container". Ensure that the external authorization factor can then be used to access an encrypted disk.

C.5 Support for Platform Power Management Modes

As indicated above, a platform on which the TOE runs may support one or more modes of "powering down" that is something less than a full shutdown by the user. In cases where these modes leave data in volatile memory, they may cause the security policies to be circumvented if the device (e.g., laptop) is taken by the attacker in this state.

Some TOEs may provide the facilities to cause information being transferred from memory to disk to be encrypted as per FDP.DSK_EXT.1.1, leaving the information correctly protected whilst the platform is in a lower power mode (and no sensitive information is maintained in-memory). In these cases, the following requirements should be used by the ST Author to specify this capability.

C.5.1 Power Management function

FDP_PM_EXT.1	**Protection of Data in Power Managed States**
FDP_PM_EXT.1.1	The TSF shall protect all data stored to the disk drive during the transition to the [assignment: powered-down state(s) for which this capability is provided] state as per FDP_DSK_EXT.1.1.
FDP_PM_EXT.1.2	On the return to a powered-on state from the state indicated in FDP_PM_EXT.1.1, the TSF shall authorize the user in the manner specified in

FIA_AUT_EXT.1.1 before any data is decrypted.

Application Note: For the first selection, the ST author fills in the state using the same name used in the Administrative Guidance for the state that is appropriately protected by the TOE.

It should be noted that it is not sufficient to use Operational Environment-based credentials to unlock the device from the indicated state; the intent is that returning from the indicated state is equivalent (from an authorization point of view) to returning from a completely powered-off state.

Assurance Activities: The evaluator shall examine the TSS to ensure that it describes the state(s) that are supported by this capability. For each state, the evaluator ensures that the TSS contains a description of how the state is entered, and the actions of the TSF on entering the state. The TSF shall also describe how the state is exited, and how the requirements are met during this transition to an operational state.

The evaluator shall check the Administrative Guidance to determine that it describes the states that are supported by the TOE, and provides information related to the correct configuration of these modes and the TOE.

The following tests must be performed by the evaluator for each supported State:

- Test 1: Following the administrative guidance, configure the Operational Environment and the TOE so that the lower power state of the platform is protected. Invoke the lower power state. On resumption of normal power, observe that an incorrect entry of the authorization factor(s) does not result in access to the system, and that correct entry of the authorization factor(s) does result in access to the system. The intent here is that the user should be first prompted for the authorization factor such that the only activity performed by the system is for the TOE to verify that the authorization factors are valid before allowing access to unencrypted data from the device.

Appendix D: Document Conventions

Except for replacing United Kingdom spelling with American spelling, the notation, formatting, and conventions used in this PP are consistent with version 3.1 of the Common Criteria (CC). Selected presentation choices are discussed here to aid the PP reader.

Selected presentation choices are discussed here to aid the PP user. The CC allows several operations to be performed on functional and assurance requirements; *refinement*, *selection*, *assignment*, and *iteration* are defined in Appendix C4 of Part 1 of the CC 3.1. Each of these operations is used in this PP.

Refinement Convention

The **refinement** operation is used to add detail to a requirement, and thus further restricts a requirement. Refinement of security requirements is denoted by the word "Refinement" in **bold text** after the element number and the additional text in the requirement in bold text. A refinement cannot "weaken" the original requirement; a TOE meeting the refined requirement must also meet the unrefined requirement in the context of the PP/ST (see appendix C.4.4 Part 1, CC 3.1). Note that a refinement may also consist of a deletion of CC words, which is indicated with ~~strikethrough~~ text.

Selection Convention

The **selection** operation is used to select one or more options provided by the CC in stating a requirement (see appendix C.4.3 Part 1, CC 3.1). Selections that have been made by the PP authors show the selection in **bold** characters, the brackets and the word "selection" removed. Selections to be filled in by the ST author are shown in square brackets with an indication that a selection is to be made, [selection:].

Assignment Convention

The **assignment** operation is used to assign a specific value to an unspecified parameter, such as the length of a passphrase (see appendix C.4.2 Part 1, CC 3.1). Showing the value in **bold** characters denotes assignments that have been made by the PP authors, the brackets and the word "assignment" are removed. Assignments to be filled in by the ST author are shown in square brackets with an indication that an assignment is to be made [assignment:].

Iteration Convention

The **iteration** operation is used when a component is repeated with varying operations (see appendix C.4.1 Part 1, CC 3.1). The iteration number (iteration_number) is show in parenthesis following the component identifier.

The iteration operation may be performed on every component. The PP/ST author performs an iteration operation by including multiple requirements based on the same component. Each iteration of a component shall be different from all other iterations of that component, which is realized by

completing assignments and selections in a different way, or by applying refinements to it in a different way. An example of an iteration is FCS_COP.1 being iterated three times to require the implementation of three different cryptographic algorithms.

Different iterations should be uniquely identified to allow clear rationales and tracings to and from these requirements.

Extended Requirement Convention

Extended requirements are permitted if the CC does not offer suitable requirements to meet the authors' needs. **Extended requirements** must be identified and are required to use the CC class/family/component model in articulating the requirements. Extended requirements will be indicated with the "EXT" inserted within the component.

Naming Convention for Assumption, Threats, Organizational Security Policies, and Objectives:

Assumptions: TOE security environment assumptions are given names beginning with "A." followed by a descriptive label all in caps (e.g., A.TRAINED_ADMINISTRATORS).

Threats: TOE security environment threats are given names beginning with "T." followed by a descriptive label all in caps (e.g., T.ACCIDENTAL_ADMIN_ERROR).

Policy Statements: Policy statements are given names beginning with "P." followed by a descriptive label all in caps (e.g., P.AUTH_FACTORS).

Security Objectives for the TOE: Security Objectives are given names beginning with "O." followed by a descriptive label all in caps (e.g., O.CRYPTOGRAPHY).

Security Objectives for the Operational Environment: Security Objectives for the operational environment are given names beginning with "OE." followed by a descriptive label all in caps (e.g., OE.NO_EVIL).

Application Notes

1 Application notes contain additional supporting information that is considered relevant or useful for the construction or use of the TOE. Application notes also contain advice relating to the permitted operations of the component.

Assurance Activities

Assurance activities serve as a Common Evaluation Methodology for the functional requirements levied on the TOE to mitigate the threat. The activities include instructions for evaluators to analyze specific aspects of the TOE as documented in the TSS, thus levying implicit requirements on the ST author to include this information in the TSS section.

Appendix E: Entropy Documentation and Assessment

The documentation of the entropy source should be detailed enough that, after reading, the evaluator will thoroughly understand the entropy source and why it can be relied upon to provide entropy. This documentation should include multiple detailed sections: design description, entropy justification, operating conditions, and health testing. This documentation is not required to be part of the TSS.

Design Description

Documentation shall include the design of the entropy source as a whole, including the interaction of all entropy source components. It will describe the operation of the entropy source to include how it works, how entropy is produced, and how unprocessed (raw) data can be obtained from within the entropy source for testing purposes. The documentation should walk through the entropy source design indicating where the random comes from, where it is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if/where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.

This design must also include a description of the content of the security boundary of the entropy source and a description of how the security boundary ensures that an adversary outside the boundary cannot affect the entropy rate.

Entropy Justification

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source exhibiting probabilistic behavior (an explanation of the probability distribution and justification for that distribution given the particular source is one way to describe this). This argument will include a description of the expected entropy rate and explain how you ensure that sufficient entropy is going into the TOE randomizer seeding process. This discussion will be part of a justification for why the entropy source can be relied upon to produce bits with entropy.

Operating Conditions

Documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. It will clearly describe the measures that have been taken in the system design to ensure the entropy source continues to operate under those conditions. Similarly, documentation shall describe the conditions under which the entropy source is known to malfunction or become inconsistent. Methods used to detect failure or degradation of the source shall be included.

Health Testing

More specifically, all entropy source health tests and their rationale will be documented. This will include a description of the health tests, the rate and conditions under which each health test is performed (e.g., at startup, continuously, or on-demand), the expected results for each health test, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.

Appendix F: Glossary of Terms

There are several definitions that apply to terms used throughout the PP:

Administrator – a user that has the capability to configure the TOE.

Authorization factor (AF) – a value submitted to the TOE to establish that the user is in the community authorized to use the hard disk and that is used (after conditioning and/or combining) as the KEK. Thus, all AFs must be successfully presented by the user since each factor is required to generate the KEK. Note that these AFs are not used to establish the particular identity of the user. An *external token authorization factor* is one that is stored on an external token. A *TPM-protected authorization factor* is one that stored on a TPM (the TPM is part of the Operational Environment, not the TOE). An *external authorization factor* is either an external token authorization factor or a TPM-protected authorization factor.

Authorized User – a user who has been provided Authorization factors by the administrator to use the TOE.

Data Encryption Key (DEK) – the key that is used by the encryption algorithm to encrypt the hard drive.

Deterministic Random Bit Generator (DRBG) – a cryptographic algorithm that produces a sequence of bits from a secret initial seed value. Without knowledge of the seed value, the output sequence should be unpredictable up to the security level of the DRBG.

Entropy Source – this cryptographic function provides a seed for a random bit generator by accumulating the outputs from one or more noise sources. The functionality includes a measure of the minimum work required to guess a given output and tests to ensure that the noise sources are operating properly.

FIPS-approved cryptographic function – a security function (e.g., cryptographic algorithm, cryptographic key management technique, or authentication technique) that is either: 1) specified in a Federal Information Processing Standard (FIPS), or 2) adopted in a FIPS and specified either in an appendix to the FIPS or in a document referenced by the FIPS.

Full Disk Encryption (FDE) -- also known as whole disk encryption, is the process of encrypting all the data on a hard drive, including the computer's OS, and permitting access to the data only after successful authentication to the FDE product. Note that software encryption products will leave a portion of the drive unencrypted for the Master Boot Record (MBR) and bootable partition. For this Protection Profile, disk encryption will be the NIST definition modified to allow software disk encryption products to leave a portion of the drive unencrypted for the MBR and bootable partition so long as no information is written there that could contain user data. If multiple drives are used, the notion of "Full Disk Encryption" requires all drives to be encrypted.

Operational environment –hardware and software that are outside the TOE boundary that supports the TOE functionality and security policy, including all hardware, associated firmware, and the operating system.

Key Encryption Key (KEK) – the key that is used to encrypt the DEK. Note: It is possible to add another layer of indirection; i.e. an intermediate key that encrypts the DEK and is encrypted by the KEK.

Keying material – the KEK, DEK, intermediate keys, authorization factors and random numbers or any other values from which a keys are derived.

Master Boot Record (MBR) – the MBR normally resides in the first sector of a hard disk. The MBR loads the bootable partition which it determines by examining the partition table.

Noise Source – the component of an RBG that contains the non-deterministic, entropy-producing activity.

Operational Environment – the environment in which the TOE is operated.

Persistent memory – data storage that retains the data long term after power is turned off.

Random Bit Generator (RBG) – a cryptographic function composed of an entropy source and DRBG that is invoked for random bits needed to produce keying material

Unauthorized User – a user that does not possess a valid authorization factor for the TOE.

Volatile memory – memory that loses its content after power is turned off.

Zeroize - this term is used to make a distinction between dereferencing a memory location and actively overwriting it with a constant. Keying material needs to be overwritten when it is no longer needed.

Appendix G: PP Identification

Tile:	Protection Profile for Software Full Disk Encryption—Mitigating the Risk of a Lost or Stolen Hard Disk
Version:	1.0
Sponsor:	National Security Agency (NSA)
CC Version:	Common Criteria for Information Technology Security Evaluation (CC) Version 3.1R3, July 2009.
Evaluation Level:	Evaluation Assurance Level (EAL) 1
Keywords:	authorization factor, authorization subsystem, DEK, disk encryption, encryption subsystem, KEK