Becrypt DISK Protect v5.2.9

Security Target

EAL2

Version 1.0

November 2009

Document History

Version	Date	Author	Description
1.0	04-Nov-09	Jussipekka Leiwo	Final

Table of Contents

List of Tables

1 Document introduction

1 This section provides preliminary information and various documenting conventions which do not formally constitute elements of a Security Target but which are used to present the Security Target to the reader, as well as other information which aims at assisting the reader in understanding the ST and the TOE it describes.

1.1 Document conventions

2 Part 2 of the Common Criteria defines an approved set of operations that may be applied to the statement of security functional requirements. Following are the approved operations and the document conventions as used within this ST to depict their application:

a) **Assignment.** The assignment operation provides the ability to specify an identified parameter within a requirement. Assignments are depicted using bolded text and are surrounded by square brackets as follows [**assignment**].

b) **Selection.** The selection operation allows the specification of one or more items from a list. Selections are depicted using bold italics text and are surrounded by square brackets as follows [***selection***].

c) **Refinement.** The refinement operation allows the addition of extra detail to a requirement. Refinements are indicated using bolded text, for **additions**, and strike-through, for ~~deletions~~.

d) **Iteration.** The iteration operation allows a component to be used more than once with varying operations. Iterations are depicted by placing a letter at the end of the component identifier as follows FDP_IFF.1a and FDP_IFF.1b.

e) **Application note.** An informal explanation by the author of the ST to highlight and explain an unusual or otherwise exceptional wording either in the requirements for an artefact of the ST or in the statement of a specific artefact in the ST.

1.2 Terminology

3 The essential terminology used in this ST is described in Table 1.

Table 1 – Terminology

Term	Description
AES	Advanced Encryption Standard, a symmetric cryptosystem defined in Federal Information Processing Standard (FIPS) Publication 197, “Advanced Encryption Standard (AES)”, 26 November 2001
Authentication data	The user-entered password and the reference password to which the user-entered password is compared during the authentication of the user.
Boot sequence	The regular sequence of processes and programs required for booting up a host PC and the operating system thereof.
Boot-loader	The TOE executable that intercepts the boot-up sequence of the host PC and injects an authentication dialogue into the sequence, and allows the boot-up to take place only upon successful user authentication

Term	Description
Cryptanalysis	An attempted attack by a threat agent to break the cryptographic protection of the TOE, or A method used by legitimate developers to assess the cryptographic strength of the TOE.
Device recovery	A specific procedure for restoring the TOE to an operational state from the Lockdown mode by a collaboration of the end user and administrator.
Disk Encryption Key	The 128-bit or 256-bit AES key used by the TOE for encrypting the hard disk of the host PC
Hibernation	A power saving feature of an operating system where the content of the RAM is written to a hard disk prior to powering off the system. The operating system can be restored to the state it was in when hibernation was invoked without a reboot.
Hibernation file	The file containing the RAM image created as part of preparation of the operating system for hibernation.
HMAC	Hashed Message Authentication Code, a technique for converting a cryptographically secure hash function into a keyed cryptographic integrity checksum
Host PC	The PC on which the TOE is installed and executed on.
Interception of a system call	The feature of a TOE in which specific system calls of the operating system running in the host PC are observed by the TOE and delayed until the processing implemented in the TOE as a reaction to that system call has been completed.
Lockdown mode	A countermeasure to a password guessing or brute force attack implemented by the TOE. Upon a number of consecutive failed authentication attempts, the TOE enters a Lockdown mode in which all accesses are denied to ensure that the attack cannot continue.
Retry counter	The data structure of the TOE keeping track of the number of consecutive failed authentication attempts, incremented each time an authentication failure occurs and reset each time an authentication is successful.
Session data	The data established upon successful authentication of the end user to allow operation of the TOE and destroyed upon termination of the session to ensure that re-authentication is required prior to the operation of the TOE can commence.
SSO (Single Sign-On)	A method of access control that allows users to enter a single set of authentication credentials (i.e. log in once) and gain access to more than one system without needing to re-enter the authentication credentials for each system separately.
Wake-up	The process of restoring a PC to the operational state from hibernation.

1.3 References

[1] Common Criteria for Information Technology Security Evaluation Part 1: Introduction and General Model, version 3.1 Revision 1, September 2006, CCMB-2006-09-001.

[2] Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, version 3.1 Revision 2, September 2007, CCMB-2007-09-002.

[3] Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, version 3.1 Revision 2, September 2007, CCMB-2007-09-003.

1.4 Document organisation

4 This document is organised into the following sections:

a) Section 1 provides introductory and preliminary explanations and document conventions to assist readers in understanding this ST.

b) The assurance families required for fulfilling assurance class ASE (ST Evaluation) at EAL2, excluding the rationales, are covered as follows:

   i) ASE_CCL.1 (Conformance claims) in Section 3.

   ii) ASE_ECD.1 (Extended components definition) In Section 6.

   iii) ASE_INT.1 (ST introduction) in Section 2.

   iv) ASE_OBJ.2 (Security objectives) in Section 5.

   v) ASE_REQ.2 (Derived security requirements) in Section 7.

   vi) ASE_SPD.1 (Security problem definition) in Section 4.

   vii) ASE_TSS.1 (TOE summary specification) in Section 8.

c) The rationales as all presented centrally in Section 9.

2 ST introduction

5 This section identifies the ST and describes the TOE in a narrative manner.

2.1 ST and TOE reference

6 The ST reference that uniquely identifies the ST is a combination of the TOE title and document version, the values of which are stated in Table 2.

Table 2 – ST identification information

ST Title	Becrypt DISK Protect v5.2.9 EAL2 Security Target
ST Version	1.0 (04-NOV-09)

7 The TOE reference is used to uniquely reference the TOE is a combination of product version, TOE Evaluation Assurance Level (EAL), and CC version identification, the values of which are stated in Table 3.

Table 3 – TOE identification information

TOE Version	Becrypt DISK Protect v5.2.9 Build 36
EAL	EAL2
CC Version Identification	Common Criteria for Information Technology (IT) Security Evaluation, Version 3.1 Revision 2 as stated in [1], [2], and [3].

2.2 TOE overview

2.2.1 Usage and major security features of the TOE

8 DISK Protect by Becrypt is a full-disk encryption product for Windows based operating systems. DISK Protect encrypts all data on a computer hard drive, including user data, the TOE executables when the TOE is not loaded and the operating system files when the operating system is not booted up.

9 The product also includes a boot-loader component. The boot-loader enforces the requirement of successful user authentication having to be completed before access is granted to the disk encryption key. Without the disk encryption key, on-the-fly encryption and decryption of the hard disk cannot commence. Consequently, the operating system and TOE files cannot be decrypted and neither the boot-up of the operating system nor the loading of the TOE can commence.

10 If the initial user authentication is successful, the hard disk becomes accessible to the boot-loader (i.e. the decryption function gets access to the necessary decryption keys) and the boot-loader can load the operating system and the TOE normally. The encryption and decryption of files happens at a lower level of disk access than on which the boot-loader operates so the boot-loader is not aware of the encryption and decryption.

11 The user authentication mechanism is password based as default, however, DISK Protect also supports secondary authentication using a USB token or smartcard device.

12 DISK Protect includes AES encryption of all data on a computer's hard drive using a 128-bit or 256-bit key and password based user authentication at the system boot time.

13 These and supportive security features the Becrypt DISK Protect implements are characterised in Table 4.

Table 4 – DISK Protect security features and characteristics

Feature	Characteristics
Full Disk Encryption	DISK Protect encrypts a computer's hard disk(s) using 128-bit or 256-bit AES data encryption. The encryption is transparent to the user. After successful authentication, the disk encryption key is released to the encryption and decryption function which automatically decrypts and encrypts all data on the fly. If the user authentication is not attempted or fails, the data encryption key remains inaccessible to the encryption and decryption function. Consequently, the data on the encrypted hard disk remains encrypted and, thus, unintelligible.
Removable media encryption	Optionally, files stored on removable media and floppy disks may be encrypted.
Multiple user support	More than a single user may gain access to a hard disk encrypted by DISK Protect. All data is encrypted with a single key but several user accounts may be created, each with a unique username-password combination, to access the key.
Single Sign-on	This optional feature synchronises the user's DISK Protect and Windows passwords to allow users to automatically log into Windows. DISK Protect can be integrated with Windows logon for both password and token based authentication. In case of smart card based authentication, the integration also provides automatic PIN entry to confirm the user's Windows certificate.
Secure hibernation	Hibernation is a mechanism to allow a computer to suppress all processing without a shut down and start up rapidly without a reboot after a lengthy period of inactivity. Hibernation is implemented by storing an image of system memory once the hibernation commences and restoring it upon activation. DISK Protect intercepts the hibernation process, encrypting the hibernation file as it is written to disk and decrypting it on start up, allowing hibernation without the risk of disclosure of user data through the hibernation files.
Device recovery	Authorised administrators can unlock the protected PCs on behalf of users who have forgotten their passwords or lost their tokens. Recovery data is gathered during the DISK Protect installation and can be used to recreate the token. The user must contact the administrators and provide the challenge code, which is then used to generate a response code that must be entered into the locked computer to regain access. Upon presenting a successful response code, the user may update the password. At no time is the user's original password exposed. Alternatively, administrators may be log in to Windows and use the Management Tool to reset the user's password.

Feature	Characteristics
Pre-boot authentication	DISK Protect can be configured to authenticate the user by password, by USB token or smart card and PIN. Authenticating the user pre-boot (i.e. prior to the commencement of the boot sequence) allows DISK Protect to encrypt the entire hard drive, including the Operating System. This ensures that the data remains inaccessible to unauthorised parties even if using low-level analysis tools. The authentication mechanism is part of the boot-loader that boots up the operating system and loads the TOE. If the pre-boot authentication fails, the boot-loader cannot access the boot files and the booting of the operating system and loading of the TOE cannot commence.
Package installation	DISK Protect may be installed and configured on individual client computers or on multiple client computers via an Installation Package. Even after installation to multiple clients, each PC must be configured with a password and a unique encryption key.
Token support	DISK Protect supports a number of smart cards and USB tokens for dual-factor authentication. Extended smart card support allows an organisation to use a card that is already part of its security systems, issuing its staff with a single card for access control and authentication.

2.2.2 TOE type

14 The TOE is not of any type defined in CC Part 1.

15 The TOE is categorised as a Data Protection product.

16 The TOE is capable of authenticating the end user and encrypting and decrypting the content of the hard disk or removable media and ensuring access to the decryption function is only available to authentic and authorised parties.

2.2.3 Hardware, software and firmware required by the TOE

17 The TOE is a software product being executed in the host PC. As such, it requires a host PC for power and connectivity. The host PC must run the operating system platform on which the TOE executes. The host PCs and operating system platforms supported are any X86 based processors running Windows XP (SP1, SP2, SP3), Windows 2000 SP4, Windows 2003 Server and Windows Vista Operating Systems.

18 The TOE mediates the file access between the operating system of the host PC and the hard disk. As such, neither the OS nor the hard drive is part of the TOE. Only the DISK Protect software that performs the mediation is within the scope of the TOE.

19 The TOE intercepts the boot sequence of the host PC and adds an authentication mechanism. This ensures that even if the host PC is compromised the TOE will maintain its ability to differentiate between legitimate and illegitimate end users and administrators.

20 Upon successful authentication of the end user, the boot-up sequence commences and the necessary OS and TOE files are decrypted to ensure successful operation of the host PC and the TOE. The actual booting of the host PC is outside the scope of the TOE.

21 The TOE also intercepts the OS system calls to hibernate and to recover from hibernation and encrypts and decrypts the hibernation file. However, the hibernation process and mechanism itself is not part of the TOE.

2.3 TOE description

22 This section describes the physical and logical scope of the TOE. The TOE architecture is such that both the physical and logical are closely interrelated.

23 The TOE consists of an initialisation module, an encryption module, an authentication module, a hibernation module, and an administrative interface. All modules and interfaces are software modules running either as background processes in the Windows operating system or as processes constituting the boot-up sequence of the TOE. As the mapping of physical modules and logical functionality is relatively straightforward, the mapping is used to describe the logical scope of the TOE in Table 5.

Table 5 – Physical and logical scope of the TOE

Physical Module	Logical functionality
Initialisation module	TOE initialisation steps are as follows: • Creation of the 128-bit or 256-bit AES key for encrypting and decrypting the hard disk. • Creation of the user-name password pair for the user of the TOE. • Setting of the value for the maximum value the retry counter may reach until the Lockdown mode is entered. • Creation of the recovery challenge distributed to the end user for presentation to the administrator in case system recovery is required.
Encryption module	The encryption module of the TOE provides the mechanism to perform encryption and decryption of the data on the hard disk. The encryption and decryption processes are transparent to the end user so no end user involvement is required once the necessary authentication stages have been successfully completed. Becrypt DISK Protect v5.2.9 also supports features for encrypting USB connected removable media.
Authentication module	The authentication module provides the interception of the boot sequence of the host PC and ensures that the TOE and OS decryption will not commence until the end user is successfully authenticated. Upon each boot-up of the host PC, the TOE inserts its own authentication screen and exchange for requesting the username and password of the end user of the TOE. Upon successful authentication of the end user, the boot sequence is allowed to commence. In case of an unsuccessful authentication, the authentication retry counter is incremented and a re-authentication is requested. If the retry counter reaches a value defined in the initialisation of the TOE the TOE enters a lockdown mode in which all end user accesses are denied. The TOE can be recovered from the lockdown mode through the administrative interface by the recovery mechanism. In addition to username-password authentication, the Becrypt DISK Protect v5.2.9 also supports token based authentication using smart cards or USB tokens and Single Sign-On (SSO).

Physical Module	Logical functionality
Hibernation module	The TOE includes a software module (hibernation module) to intercept the system hibernation and wake-up calls. This resident software intercepts the hibernation command and upon sensing the hibernation, suspends the hibernation until the hibernation file is created. The TOE then encrypts the hibernation file and resumes the hibernation. Upon a system wake-up, the hibernation module receives a broadcast message indicating the system is about to wake up from hibernation and upon receiving that message, decrypts the hibernation file so that the wake-up from hibernation may commence.
Administrative Interface	The administrative interface of the TOE allows the recovery of the TOE by providing an interface for the response to the recovery challenge the TOE administrator may compute from the recovery challenge.

3 Conformance claims

24 The following conformance claims are made for the TOE and ST:

a) **CCv3.1 Rev.2 conformant**. The ST and the TOE are Common Criteria conformant to Common Criteria version 3.1 Revision 2 defined in [1], [2] and [3].

b) **Part 2 conformant.** The ST is Common Criteria Part 2 conformant.

c) **Part 3 conformant.** The ST is Common Criteria Part 3 conformant.

d) **Package conformant.** The ST is package conformant to the package Evaluation Assurance Level EAL2 as defined in [3].

e) **Protection Profile conformance**. The ST does claims conformance to the following Protection Profiles: **None**.

4 Security problem definition

25 The TOE is concerned with the protection of the following assets enumerated in Table 6.

Table 6 – Assets protected by the TOE

Identifier	Asset statement
AST.HIB_FILE	Confidentiality during hibernation of the hibernation file created by the operating system of the host PC upon hibernation of the host PC.
AST.HD_DATA	Confidentiality of the user data and operating system configuration files and integrity of the TOE executables and operating system data when stored on the hard disk of the host PC.
AST.LEG_ACC	Legitimacy of access to the data on the hard disk of the host PC.

26 The subjects, some of which constitute threat agents as highlighted in the description of threats, are stated in Table 7.

Table 7 – Subjects relevant to the TOE

Identifier	Subject definition
S.END_USER	The legitimate end user of the TOE and the host PC.
S.ADMINISTRATOR	The legitimate administrator of the TOE and the host PC.
S.ANYBODY	Human user other than S.END_USER or S.ADMINISTRATOR.
S.ANY_SW	Any software, either analytics software used by S.ANYBODY or potentially hostile software residing and being executed in the host PC or in any other PC into which the hard disk of the host PC may be transferred to while the TOE is shut down or hibernating.

4.1 Threats

27 Threats enumerated in Table 8 are relevant to the TOE.

Table 8 – Threat statements

Identifier	Threat statement
T.HIB_FILE	S.ANY_SW succeeds in violating AST.HIB_FILE by recovering from the hard disk and restoring the original hibernation file so that the host PC or a clone of the host PC with the original or replicated hard disk of the legitimate host PC can be woken up from hibernation or the state of the applications active during the hibernation of the host PC can be deduced without the presence of S.END_USER or S.ADMINISTRATOR. Once the host PC hibernates, the OS generates a hibernation file that contains the image of the OS and running applications at the time that the hibernation occurred. As the OS does not shut down during hibernation, the TOE remains active and shall be recovered without further authentication. Subsequently, access to the hibernation file would give the attacker access to sensitive information stored on the HD of the host.

T.HD_DATA	S.ANY_SW or S.ANYBODY succeeds in violating AST.HD_DATA by detaching the hard disk from the host PC and subjecting it to cryptanalysis and other hardware level penetration attempts. The detachment of the hard disk may take place at any time the TOE and the host PC are not powered on. Obviously, the host PC cannot be expected reside in a secure premises. If it did, it would be trusted and there would be no need for additional protection. Consequently, it is possible for any malicious parties to gain physical access to the host PC when not powered on (i.e. the end user of the TOE is not in the physical vicinity of the host PC) and subject it to cryptanalysis or any other low-level penetration attempts. This threat concerns with any such attacks that may occur possibly in an environment different from the regular environment of the host PC when the host PC of the TOE is powered down and the TOE is not active.
T.UNAUTH	S.ANYBODY succeeds in masquerading as S.END_USER by correctly guessing the password used for end user authentication. The most straightforward way for an illegitimate user to attempt to gain access to the TOE and the user data or OS files stored on the hard disk of the host PC is to engage in password guessing or brute force attacks on the username-password mechanism of the TOE. Any correct guess would cause a loss of TOE’s ability to differentiate between legitimate and illegitimate end users and render the TOE unable to ensure that access to the protected data is only granted to legitimate end users. Password guessing and brute force attacks may exploit two potential weaknesses in password based authentication mechanisms: the low quality of passwords and the high number of authentication attempts allowed to the end users. If the passwords are not of high quality and the number of guessing attempts is high, the probability of the correct guess during the password guessing attack and the probability of discovery of the correct password by brute force attack through the authentication interface may become unacceptably high.
T.RECOVERY	S.ANYBODY succeeds in bypassing the restrictions in recovering the TOE and without participation of the end user and administrator succeeds in recovering the TOE that has entered a Lockdown mode as a result of assumed password guessing attack. The administrative interface of the TOE includes a functionality to recover the TOE from the Lockdown mode. Both the legitimate end user and the legitimate administrator are required to participate in the recovery. As the TOE enters a Lockdown mode once the retry counter reaches the predefined number of allowed authentication attempts, the only way to succeed in a password guessing attack is either to succeed in the guessing before the TOE enters the Lockdown mode (which is highly unlikely) or to succeed in restoring the TOE without invoking the legitimate TOE recovery function.

4.2 Organisational security policies

28 The organisational security policies enumerated in Table 9 are relevant to the TOE.

Table 9 – Organisational Security Policies

Identifier	OSP statement
OSP.OS_CONF	The operating system of the host PC is configured to be secure. Of particular concern is the prevention of end users from having administrative rights. Furthermore, the credentials for entering an administrative role within the OS are sufficiently secure to prevent with a high likelihood unauthorised, potentially malicious users from succeeding in becoming administrators.

4.3 Assumptions

29 The Assumptions enumerated in Table 10 Assumptions are relevant to the TOE.

Table 10 Assumptions

Identifier	Assumption statement
A.PASSWORD	The end users of the TOE are aware of the importance of the quality of passwords to the security of the TOE and the sensitive data residing on the hard disk of the host PC and only select passwords of good quality when initialising the TOE. End users also keep the passwords secret and do not write them down or disclose them to any other system or user.

5 Security objectives

30 This section states the exact security objectives for the TOE so that the security problem definition is adequately and completely addressed. The security objectives are stated for the TOE and for the operational environment of the TOE.

5.1 Security objectives for the TOE

31 Security objectives for the TOE are enumerated in Table 11.

Table 11 – Security objectives for the TOE

Identifier	Objective statement
O.HIB_FILE	The hibernation file is protected against unauthorised disclosure while the TOE is hibernating.
O.HD_DATA	The user data and the OS configuration files stored on the hard disk are protected against unauthorised disclosure, and the OS files and TOE executables against unauthorised modification.
O.AUTH	The TOE has a means to differentiate between legitimate and illegitimate end users and only grants access to the protected data to legitimate users. The probability of an illegitimate user gaining access to the TOE is negligibly low.
O.ADMIN	Access to the TOE recovery function is only granted to legitimate administrator and legitimate end user when collaborating.

5.2 Security objectives for the environment

5.2.1 Security objectives for the IT environment

32 Security objectives for the IT environment of the TOE are stated in Table 12.

Table 12 – Security objectives for the IT environment of the TOE

Identifier	Objective statement
OE.PASSWORD	The passwords accepted by the host PC for use by the TOE are of high quality.

5.2.2 Security objectives for the non-IT environment

33 The following security objectives are stated for the non-IT environment of the TOE.

Identifier	Objective statement
ON.PASSWORD	The end users keep their passwords secret and do not write them down or disclose them to any party other than the host PC during the password selection.

6 Extended components definition

34 The following extended components are defined for the TOE: **None**.

35 There are no extended components applicable to the TOE. Hence, none of the requirements for the Extended Components Definition (ASE_ECD) are applicable to this ST and shall be omitted.

7 IT security requirements

7.1 Overview

36 This section defines the security requirements satisfied by the TOE. Each requirement has been extracted from version 3.1 of the Common Criteria, part 2 providing functional requirements and part 3 providing assurance requirements.

7.2 TOE security functional requirements

37 This section contains the security functional components from part 2 of the Common Criteria with the operations completed.

38 Standard Common Criteria text is in regular black font and the text inserted to perform an operation on the requirement is in accordance with the conventions specified in section 1 of this ST.

Table 13 – Summary of TOE security functional requirements

Identifier	Title
Cryptographic support	
FCS_CKM.1	Cryptographic key generation
FCS_CKM.4	Cryptographic key destruction
FCS_COP.1a	Cryptographic operation (Hibernation file)
FCS_COP.1b	Cryptographic operation (Hard disk)
User data protection	
FDP_RIP.1a	Subset residual information protection (TOE Data)
FDP_RIP.1b	Subset residual information protection (Cryptographic Keys)
Identification and authentication	
FIA_AFL.1	Authentication failure handling
FIA_ATD.1	User attribute definition
FIA_UAU.1	Timing of authentication
FIA_UID.1	Timing of identification
Security management	
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security roles
Protection of the TSF	
FPT_FLS.1	Failure with preservation of secure state

Identifier	**Title**
FPT_TST.1a	TSF testing (Key generation)
FPT_TST.1b	TSF testing (Start-up)
Trusted Paths/Channels	
FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1	Trusted path

7.2.1 Cryptographic support

7.2.1.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to:	No other components.
FCS_CKM.1.1	The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**AES**] and specified cryptographic key sizes [**128 bits and 256 bits**] that meet the following: [**Federal Information Processing Standard (FIPS) Publication 197, "Advanced Encryption Standard (AES)", 26 November 2001**].
Dependencies:	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction
Notes:	None.

7.2.1.2 FCS_CKM.4 Cryptographic key destruction

Hierarchical to:	No other components.
FCS_CKM.4.1	The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [**Zeroization**] that meets the following: [**FIPS 140-2 Level 1**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]
Notes:	The administrator can zeroise the key. Alternatively, the key is destroyed upon device re-formatting prior to re-initialisation.

7.2.1.3 FCS_COP.1a Cryptographic operation (Hibernation file)

Hierarchical to:	No other components.
FCS_COP.1a.1	The TSF shall perform [a) **Encryption of the hibernation file once the host PC of the TOE hibernates, and** b) **Decryption of the hibernation file once the host PC of the TOE recovers from hibernation.**] in accordance with a specified cryptographic algorithm [**AES**] and cryptographic key sizes [**128 bits and 256 bits**] that meet the following: [**Federal Information Processing Standard (FIPS) Publication 197, "Advanced Encryption Standard (AES)", 26 November 2001**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]

	FCS_CKM.4 Cryptographic key destruction
Notes:	None.

7.2.1.4 FCS_COP.1b Cryptographic operation (Hard disk)

Hierarchical to:	No other components.
FCS_COP.1b.1	The TSF shall perform [**a) Decryption of the data requested from the hard disk, and** **b) Encryption of the data forwarded to the hard disk.**] in accordance with a specified cryptographic algorithm [**AES**] and cryptographic key sizes [**128 bits and 256 bits**] that meet the following: [**Federal Information Processing Standard (FIPS) Publication 197, "Advanced Encryption Standard (AES)", 26 November 2001**].
Dependencies:	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction
Notes:	This function takes place immediately after the boot-up authentication has been successfully completed. Therefore, the SFR covers also the operating system files and TOE executables (other than the boot-loader) that the boot-loader loads during the operating system boot-up and TOE start-up.

7.2.2 User data protection

7.2.2.1 FDP_RIP.1a Subset residual information protection (TOE data)

Hierarchical to:	No other components.
FDP_RIP.1a.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***deallocation of the resource from***] the following objects: [**a) Authentication Data, and** **b) Session Data.**].
Dependencies:	None.
Notes:	None.

7.2.2.2 FDP_RIP.1b Subset residual information protection (Cryptographic keys)

Hierarchical to:	No other components.
FDP_RIP.1b.1	The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] the following objects: [**AES Key**].
Dependencies:	None.
Notes:	None.

7.2.3 Identification and authentication

7.2.3.1 FIA_AFL.1 Authentication failure handling

Hierarchical to:	No other components.
FIA_AFL.1.1	The TSF shall detect when [***an administrator configurable positive integer within*** [**a range of values no more than three (3)**]] unsuccessful authentication attempts occur related to [**User authentication**].
FIA_AFL.1.2	When the defined number of unsuccessful authentication attempts has been [***surpassed***], the TSF shall [**Enter a Lockdown mode**].
Dependencies:	FIA_UAU.1 Timing of authentication
Notes:	None.

7.2.3.2 FIA_ATD.1 User attribute definition

Hierarchical to:	No components.
FIA_ATD.1.1	The TSF shall maintain the following list of security attributes belonging to individual **TOE** users: [**Role**].
Dependencies:	None.
Notes:	None.

7.2.3.3 FIA_UAU.1 Timing of authentication

Hierarchical to:	No other components.
FIA_UAU.1.1	The TSF shall allow [**a) Device Initiation, and** **b) Entering a Lockdown Mode.**] on behalf of the user to be performed before the user is authenticated.
FIA_UAU.1.2	The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	FIA_UID.1 Timing of identification
Notes:	None.

7.2.3.4 FIA_UID.1 Timing of identification

Hierarchical to:	No other components.
FIA_UID.1.1	The TSF shall allow [**a) Initiation of a trusted path for end user authentication, and** **b) Initiation of a trusted channel for TOE recovery.**

	] on behalf of the user to be performed before the user is identified.
FIA_UID.1.2	The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.
Dependencies:	None.
Notes:	None.

7.2.4 Security Management

7.2.4.1 FMT_SMF.1 Security Management

Hierarchical to:	No other components.
FMT_SMF.1.1	The TSF shall be capable of performing the following security management functions: [a) **Manage users,** b) **Execute TOE recovery,** c) **Set local machine policy,** d) **View encryption status (show status),** e) **View product information (show status),** f) **Configure fixed disk encryption (generate encryption key),** g) **Add users,** h) **Power up self tests + conditional test, and** i) **Trigger zeroization of cryptographic keys.**].
Dependencies:	None.
Notes:	None.

7.2.4.2 FMT_SMR.1 Security roles

Hierarchical to:	No other components.
FMT_SMR.1.1	The TSF shall maintain the roles [a) **Administrator, and** b) **End User.**].
FMT_SMR.1.2	The TSF shall be able to associate users with roles.
Dependencies:	FIA_UID.1 Timing of identification
Notes:	None.

7.2.5 Protection of the TSF

7.2.5.1 FPT_FLS.1 Failure with Preservation of Secure State

Hierarchical to:	No other component
FPT_FLS.1.1	The TSF shall preserve a secure state when the following types of failures occur: [a) **Power failure, and** b) **Hibernation of the host PC.**].
Dependencies:	None.
Notes:	None.

7.2.5.2 FPT_TST.1a TSF Testing (Key generation)

Hierarchical to:	No other components.
FPT_TST.1a.1	The TSF shall run a suite of self tests [***at the conditions*** [**Prior to the generation of cryptographic keys**]] to demonstrate the correct operation of [**The key generation function**].
FPT_TST.1a.2	The TSF shall provide authorised users with the capability to verify the integrity of [**the Random Number Generator**].
FPT_TST.1a.3	The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.
Dependencies:	None.
Notes:	None.

7.2.5.3 FPT_TST.1b TSF Testing (Start-up)

Hierarchical to:	No other components.
FPT_TST.1b.1	The TSF shall run a suite of self tests [***during initial start-up***] to demonstrate the correct operation of [**the TSF**].
FPT_TST.1b.2	The TSF shall provide authorised users with the capability to verify the integrity of [a) **Cryptographic algorithms and** b) **TOE files.**].
FPT_TST.1b.3	The TSF shall provide authorised users with the capability to verify the integrity of stored TSF executable code.
Dependencies:	None.

Notes:	The authorised user can verify the integrity of the TSF executable code by rebooting the TOE. Self tests are performed on the cryptographic library. The library is comprised of two sub components, viz., a 16-bit sub component object that is designed to operate in a pre-OS or DOS environment and a 32-bit sub component object designed to operate in 32-bit operating environments. Cryptographic algorithms are tested by Known Answer Tests (KAT) and the tested cryptographic algorithms are used for testing the integrity of the TOE files. The TOE only includes AES as a SFR as that is the essence of the protection of the user data by the TOE but also SHA-256 and HMAC-SHA256 are applied in the testing. Failure of any of the KAT tests or integrity tests will halt the boot of the TOE. If all the tests pass, the TOE shall complete the boot sequence and resume into a normal mode.

7.2.6 Trusted Paths/Channels

7.2.6.1 FTP_ITC.1 Inter-TSF Trusted Channel

Hierarchical to:	No other components.
FTP_ITC.1.1	The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.
FTP_ITC.1.2	The TSF shall permit [***another trusted IT product***] to initiate communication via the trusted channel.
FTP_ITC.1.3	The TSF shall initiate communication via the trusted channel for [**Recovery of the TOE**].
Dependencies:	None.
Notes:	None.

7.2.6.2 FTP_TRP.1 Trusted path

Hierarchical to:	No other components.
FTP_TRP.1.1	The TSF shall provide a communication path between itself and [***local***] users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [***modification and disclosure***].
FTP_TRP.1.2	The TSF shall permit [***the TSF***] to initiate communication via the trusted path.
FTP_TRP.1.3	The TSF shall require the use of the trusted path for [***initial user authentication***].
Dependencies:	None.
Notes:	None.

7.3 TOE security assurance requirements

39 The assurance package for the evaluation of the TOE is Evaluation Assurance Level 2 (EAL2).

40 EAL2 assurance requirements provide confidence in the security functionality of the TOE by analysis using a functional and interface specification, guidance documentation and the high-level design of the TOE, to understand the security behaviour.

41 The analysis is supported by independent testing of the TOE security functions, evidence of developer testing based on the functional specification, selective independent confirmation of the developer test results, strength of function analysis, and evidence of a developer search for obvious vulnerabilities.

42 EAL2 also provides assurance through a configuration list for the TOE, and evidence of secure delivery procedures.

43 Table 14 lists the TOE security assurance requirements for this evaluation. Complete details of all assurance components are located in part 3 of the Common Criteria.

Table 14 – Summary of TOE security assurance requirements

Assurance class	Assurance components
Development (ADV)	ADV_ARC.1
	ADV_FSP.2
	ADV_TDS.1
Guidance Documents (AGD)	AGD_OPE.1
	AGD_PRE.1
Life-Cycle Support (ALC)	ALC_CMC.2
	ALC_CMS.2
	ALC_DEL.1
Security Target Evaluation	ASE_CCL.1
	ASE_ECD.1
	ASE_INT.1
	ASE_OBJ.2
	ASE_REQ.2
	ASE_SPD.1
	ASE_TSS.1
Tests (ATE)	ATE_COV.1
	ATE_FUN.1

Assurance class	Assurance components
	ATE_IND.2
Vulnerability Assessments (AVA)	AVA_VAN.2

8 TOE Summary Specification

8.1 Overview

44 This chapter provides the TOE summary specification, a high-level definition of the security functions of the TOE and a summary of how those Security Functions meet the SFR's.

8.2 Security functions

45 The TOE security functions include the following:

a) **Data protection.** The TOE provides the capability to protect data in storage by encrypting when stored and decrypting when retrieved from storage and passed to the host PC.

b) **User authentication.** The TOE provides the capability to authenticate users and ensure access to the protected files is only granted to legitimate users.

c) **Security boot-up.** The TOE provides the capability to ensure that both the TOE and the host PC boot into a secure state and violation of integrity of the TOE files or the OS files is detected and the booting halted.

46 Each of the security functions listed above is discussed in more detail below. The relationship of the security functions and the SFRs for the TOE is illustrated in Table 15[1].

Table 15 – Security functions and SFRs

Security function	Applicable SFRs
HD data protection	FCS_CKM.1 Cryptographic key generation FCS_CKM.4 Cryptographic key destruction FCS_COP.1a Cryptographic operation (Hibernation file) FCS_COP.1b Cryptographic operation (Hard disk) FDP_RIP.1b Subset residual information protection (Cryptographic Keys) FPT_TST.1a TSF testing (key generation)

[1] **Application note**: A TOE such as the one described in this ST can be characterised by a relatively small number of functionalities all of which are highly interrelated. As is illustrated in the table, many SFRs for the TOE contribute to more than one security function and also, as shall be demonstrated later in the Rationale, for more than a single security objective. This is not a violation of good security design principles but an inherent feature on a TOE such as the one described in this ST.

Security function	Applicable SFRs
User authentication	FIA_AFL.1 Authentication failure handling FIA_ATD.1 User attribute definition FIA_UAU.1 Timing of authentication FIA_UID.1 Timing of identification FMT_SMF.1 Specification of Management Functions FMT_SMR.1 Security roles FTP_ITC.1 Inter-TSF trusted channel FTP_TRP.1 Trusted path
Secure boot-up	FDP_RIP.1a Subset residual information protection (TOE Data) FPT_FLS.1 Failure with preservation of secure state FPT_TST.1b TSF testing (Start-up)

8.2.1 Data protection

47 Protection of the data on the hard disk of the host PC is the essential cryptographic feature of the TOE. The TOE includes a library of cryptographic functions that is used for implementing the range of cryptographic functions constituting the security function HD data protection of the TOE.

48 The TOE includes a random number generator and a key generation function for generating the 128-bit and 256-bit AES key used as a disk encryption key. This covers FCS_CKM.1. When disk encryption keys are generated, specific measures are implemented to ensure that any residual information from the data structures possibly holding the previous disk encryption key is fully erased. This covers FDP_RIP.1b.

49 The TOE administrator also has access to a function for cryptographically secure erasure of those keys by zeroization. This covers FCS_CKM.4.

50 Prior to the use of the data encryption keys, during the start-up of the TOE, a number of known answer tests are implemented to ensure the integrity of the cryptographic functions. If an integrity violation is detected (i.e. any of the known answer tests fails), the boot-up is halted. This covers FPT_TST.1a[2].

51 If the TOE is successfully booted, all data stored on the hard disk of the host PC is encrypted by the TOE when stored and decrypted by the TOE when retrieved. This covers FCS_COP.1b.

52 If the host PC of the TOE enters hibernation, the TOE intercepts the hibernation system call and halts it until the hibernation file is encrypted. Upon wake-up, the TOE decrypts the hibernation file so that the wake-up may proceed. This covers FCS_COP.1a.

8.2.2 User authentication

53 For the initial user authentication, the TOE provides a dedicated authentication window and exchange that ensures that the authentication exchange remains confidential even if the host PC is compromised. This covers FTP_TRP.1.

[2] **Application note**: the boot-up sequence of the TOE is relatively complex and includes a number of stages, each of which implements a subset of the self tests. The detailed description of the sequence and self tests relevant to different stages are described in the subsequent documentation of the TOE.

54 Identification and authentication of users is required for each action except those explicitly stated as not requiring identification (FIA_UID.1) or authentication (FIA_UAU.1).

55 User authentication is username and password based. To prevent password guessing attacks, the TOE maintains a counter of consecutive authentication failures, called retry counter. If the counter value exceeds a threshold set by the TOE administrator, the TOE enters a Lockdown mode. This covers FIA_AFL.1.

56 If the TOE enters a Lockdown mode, all accesses are denied and the TOE can only be restored by the collaboration of the end user and administrator who must establish a trusted channel between the admin console and the TOE and use that channel to execute TOE recovery. This covers FTP_ITC.1.

57 Upon successful authentication, each user is assigned a dedicated role (FIA_ATD.1). The roles are end user or administrator (FMT_SMR.1), and the TOE ensures that management functions are only available to the users entering an administrative role of the TOE (FMT_SMF.1).

8.2.3 Secure boot-up

58 The secure boot-up of the TOE consists of three main features:

a) The TOE performs a self test on the cryptographic algorithms in Pre-OS crypto library sub component. The self test includes checking the HMAC codes and digital signatures for the TOE files during the initialisation and verifies the HMAC codes and digital signatures prior to the commencement of the relevant stages of the boot-up procedure. If any of the integrity checks fails, TOE halts the boot-up. Once the stages in the boot-up commence, the TOE ensures that the authentication data and any other sensitive data is sufficiently cleared when no longer necessary. This is to ensure that the residual information is not available to the potentially hostile software running on the host PC. This covers FPT_TST.1b and FDP_RIP.1a.

b) While the boot-up commences, additional self tests are executed on 32-bit sub component of the TOE cryptographic library to ensure that the critical components function correctly. This covers FPT_TST.1b.

c) If any of the checks fails, the TOE ensures that no insecure state results from the failure and halts the boot-up to ensure that the TOE or the host PC shall not boot into an insecure state. Additionally, if the TOE enters a hibernation mode after the boot-up, measures are taken to ensure that the hibernation file is sufficiently protected prior to the hibernation takes place. These features cover FPT_FLS.1.

9 Rationale

9.1 Conformance claim rationale

59 The Conformance Claim of this ST does not claim conformance to any Protection Profile. Hence, there are no elements to be covered in the conformance claim rationale.

9.2 Security objectives rationale

60 Security objectives rationale is provided for the TOE and for the environment of the TOE.

9.2.1 Security objectives for the TOE

61 Table 16 provides a mapping of the TOE Security objectives and threats and a justification for the mapping.

Table 16 – Mapping of TOE security objectives to threats

Threats	Objective	Justification
T.HIB_FILE	O.HIB_FILE	T.HIB_FILE concerns with the protection of the hibernation file when the host PC of the TOE enters hibernation. During hibernation, malicious parties may attempt to use crypto analytical or other low level tools to deduce sensitive data protected by the TOE from the hibernation file. In order to prevent such analysis, the hibernation file must be protected so that the attempts fail with an overwhelming probability. Yet, the host PC must be able to successfully recover from the hibernation into a secure state upon receiving a wake-up signal from the operating system of the host PC. O.HIB_FILE concerns with the protection of the hibernation file when the host PC enters hibernation and the recovery of the hibernation file at the wake-up. The content of the hibernation file should not be made available to any party prior to the receipt of the wake-up signal from the operating system of the host PC. Enforcing O.HIB_FILE therefore fully covers T.HIB_FILE.
T.HD_DATA	O.HD_DATA	T.HD_DATA concerns with the protection of the data stored on the hard disk of the host PC. This data includes user data, OS data and TOE executable files. O.HD_DATA concerns with the protection of the data stored on the hard disk so that only on behalf of legitimate parties (i.e. those granted access to the decryption keys) shall the hard disk be decrypted. Without access to the decryption, all parties are denied from access to the data on the hard disk. Therefore, enforcing O.HD_DATA fully covers T.HD_DATA.

Threats	Objective	Justification
T.UNAUTH	O.HD_DATA O.AUTH	T.UNAUTH concerns with the prevention of unauthorised access to the data protected by the TOE when stored on the hard disk of the host PC. This has to be addressed through two concerns: First, the data on the hard disk must be protected so that the probability of an unauthorised party succeeding in deducing the data without proper authorisation (i.e. access to the functions of the TOE) is overwhelmingly low. Second, the data on the hard disk of the host PC must be protected so that the threat agents cannot succeed in attempts to bypass the authentication features of the TOE and that the probability of correctly guessing the password of the legitimate end user is overwhelmingly low[3]. The inability of an unauthentic party to engage in crypto analytical or other attacks using low level tools to deduce the content of the protected data is covered by O.HD_DATA. The authenticity of users and the low likelihood of an unauthentic party to correctly guess the correct password are covered by O.AUTH. Jointly, O.HD_DATA and O.AUTH fully cover T.UNAUTH.
T.RECOVERY	O.ADMIN OSP.OS_CONF	T.RECOVERY concerns with the ability of an unauthentic party to recover the TOE from the Lockdown mode. As the measures to prevent T.UNAUTH from occurring enforce the Lockdown mode as a response to a password guessing or brute force attack, the TOE must also ensure that once it enters the Lockdown mode, only legitimate end user, when collaborating with a legitimate administrator, can recover the TOE. Preserving O.ADMIN ensures that the joint effort of the legitimate end user and the legitimate administrator of the TOE is required to recover the TOE from the lockdown mode and that no administrative functions are available to the end user of the TOE so that the recovery could be triggered without the presence of the administrator. The administrator also requires administrative access to the host PC to fully execute the recovery from the Lockdown mode. As the host PC and the operating system thereof are outside the scope of the TOE, it is essential that there exists a policy stating the administrative regulations on the host PC to ensure that only legitimate administrative may gain administrative access to the PC. Consequently, preservation O.ADMIN and enforcement of OSP.HOST_PC jointly do fully addresses T.RECOVERY.

[3] **Application note**: Passwords are imported from the host PC which enforces the quality controls. If the password is at least 6 ASCII characters of length and the maximum value of a retry counter is 3, the success probability of guessing the correct password will be sufficiently low.

9.2.2 Security objectives for the environment

62 Table 17 provides a mapping of the Security objectives for the environment of the TOE to relevant threats, assumptions and organisational security policies, as well as a justification for the mapping. There are no assumptions governing the usage and operation of the TOE, hence no assumptions are relevant to the mapping and justification.

Table 17 – Mapping of security objectives for the environment to threats, assumptions and OSPs

Threat/assumption/OSP	Objective	Justification
A.PASSWORD	OE.PASSWORD ON.PASSWORD	The passwords used to authenticate legitimate users to the TOE are generated in the host PC during the initialisation of the TOE. The administrator of the TOE and the administrator of the host PC of the TOE may set the criteria for the quality of passwords through the configuration options available within the host PC. Consequently, there are no functional measures for the TOE to control the quality of passwords. Consequently, a policy must exist and be enforced to state the minimum quality requirements the passwords forwarded to the TOE must meet. Once the passwords are created, the end users must keep them secret, not write them down and not disclose to any user or computer system other than the TOE. This ensures that the attackers must indeed attempt to guess the correct password if they intend to attempt password guessing attack. Preserving OE.PASSWORD and ON.PASSWORD ensures that the assumptions about the quality and management of passwords, A.PASSWORD, are fully covered by the environment of the TOE.

9.3 Security requirements rationale

63 SFR dependency rationaleTable 18 demonstrates the mutual supportiveness of the SFR's for the TOE by demonstrating how the SFR dependencies are fulfilled by the TOE, or justifying those dependencies not implemented.

64 The SARs relevant to the TOE constitute an evaluation assurance level EAL2 as defined in Common Criteria and include no extensions or augmentations. Therefore, as a complete evaluation assurance level, they are a mutually supportive set and require no further justification.

Table 18 – TOE SFR dependency demonstration

SFR	Dependency	Justification
FCS_CKM.1	[FCS_CKM.2 Cryptographic key distribution, or FCS_COP.1 Cryptographic operation] FCS_CKM.4 Cryptographic key destruction	FCS_COP.1a by the TOE FCS_COP.1b by the TOE FCS_CKM.4 by the TOE
FCS_CKM.4	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation]	FCS_CKM.1 by the TOE
FCS_COP.1a	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 by the TOE FCS_CKM.4 by the TOE
FCS_COP.1b	[FDP_ITC.1 Import of user data without security attributes, or FDP_ITC.2 Import of user data with security attributes, or FCS_CKM.1 Cryptographic key generation] FCS_CKM.4 Cryptographic key destruction	FCS_CKM.1 by the TOE FCS_CKM.4 by the TOE.
FDP_RIP.1a	None.	None.
FDP_RIP.1b	None.	None.
FIA_AFL.1	FIA_UAU.1 Timing of authentication	FIA_UAU.1 by the TOE
FIA_ATD.1	None.	None.
FIA_UAU.1	FIA_UID.1 Timing of identification	FIA_UID.1 by the TOE
FIA_UID.1	None.	None.
FMT_SMR.1	FIA_UID.1 Timing of identification	FIA_UID.1 by the TOE

SFR	Dependency	Justification
FMT_SMF.1	None.	None.
FPT_FLS.1	None.	None.
FPT_TST.1a	None.	None.
FPT_TST.1b	None.	None.
FTP_ITC.1	None.	None.
FTP_TRP.1	None.	None.

9.3.1 Tracing of SFR to security objectives

65 Table 19 provides the mapping of the TOE SFRs and the security objectives for the TOE.

Table 19 – Mapping TOE SFRs to objectives

Objective	SFRs	Demonstration
O.HIB_FILE	FCS_CKM.1 FCS_CKM.4 FCS_COP.1a FDP_RIP.1a FPT_FLS.1 FPT_TST.1a	O.HIB_FILE is preserved if a) the cryptographic keys used as disk encryption keys for encrypting the hibernation file are of good quality, b) the cryptographic keys are securely destroyed when no longer needed, c) the actual encryption and decryption of the hibernation file is appropriate, and d) the encryption and decryption of the hibernation file is invoked at each hibernation and wake-up. Quality of the cryptographic keys concerns with the generation of the disk encryption keys. The TOE implements a random number generator and a cryptographic key creation function which is used for creating the data encryption key. The data encryption key is a 128-bit or 256-bit AES key. This same key is used both for encrypting and decrypting the data on the hard disk of the host PC and for encrypting the hibernation file once the host PC hibernates and for decrypting the encrypted hibernation file once the host PC wakes up from hibernation. Prior to the generation of the data encryption key, the TOE implements various self tests to ensure that the random number generation functions correctly so that the resulting cryptographic keys are indeed random. The key generation part of O.HIB_FILE is fulfilled by FCS_CKM.1 and FPT_TST.1a. Key destruction concerns with the provision of a zeroization function which destroys the keys. Also, upon re-generation of the key, the TOE must ensure that any information about the previous key is wiped from the data structures to ensure that each key is fully unique. The key destruction part of O.HIB_FILE is fulfilled by FCS_CKM.4 and FDP_RIP.1a. Upon hibernation (i.e. Hibernation of the host PC as defined in FPT_FLS.1), the hibernation file stored on the hard disk by the operating system of the host PC is encrypted when stored on the hard disk prior to the hibernation of the host PC and decrypted when the TOE detects that the host PC is broadcasting a signal for a wake-up from the hibernation. The TOE achieves this by intercepting the hibernation and wake-up signals of the operating system and implementing the necessary security processing prior to allowing the hibernation of wake-up to commence. This feature covers the encryption and decryption of the hibernation file and is fulfilled by FCS_COP.1a. Finally, to ensure that the hibernation file is encrypted at each hibernation and decrypted at each wake up, the TOE treats hibernation of a host PC as a failure under which a secure state must be maintained. The hibernation signal is interpreted as a failure under which the TOE must be protected and the wake-up as a signal that the failure has been recovered from. The means to ensure a secure state during the failure, i.e. hibernation, is implemented by the

Objective	SFRs	Demonstration
		encryption of the hibernation file so that unauthorised parties can not gain access to the content of the file. Alternatively, the Host PC may encounter a power failure as defined in FPT_FLS.1. This may occur either accidentally through a power failure or as an intentional but uncontrolled shut-down of the host PC. In this case the hibernation does not take place as the TOE depends on the host PC for power and once the host PC looses power, the TOE also looses power immediately. Therefore, there is also no need to protect the hibernation file as the host PC does not create it. Instead, the TOE protects the authentication state which may remain set. This protection takes place through the controlled boot-up of the TOE as discussed in O.AUTH. It is also noted that hibernation may occur due to a number of reasons. Some of these reasons may look like power failures. On laptop PC's acting as Host PC, for example, low battery state may trigger hibernation. However, at the signals level all this is invisible to the TOE. To the TOE, hibernation is just a sequence of hibernation and wake-up calls no matter what triggered those calls. Power failure as defined in FPT_FLS.1, on the other hand, is an abrupt loss of power that causes loss of power of the TOE so that hibernation cannot take place.
O.HD_DATA	FCS_CKM.1 FCS_CKM.4 FCS_COP.1b FDP_RIP.1a FPT_TST.1a	To fulfil O.HD_DATA, the TOE must address a) the generation of good quality cryptographic keys, b) destruction of those keys when no longer needed, and c) the actual encryption and decryption of the data stored on the hard disk. Quality of the cryptographic keys concerns with the generation of the keys. The TOE implements a random number generation and cryptographic key creation function which is used for creating the data encryption key. The data encryption key is a 128-bit or 256-bit AES key. This same key is used both for encrypting and decrypting the data on the hard disk of the host PC and for encrypting/decrypting the hibernation file. Prior to the generation of the data encryption key, the TOE implements various self tests to ensure that the random number generation functions correctly so that the resulting cryptographic keys are indeed random. The key generation part of O.HD_DATA is fulfilled by FCS_CKM.1 and FPT_TST.1a. Key destruction concerns with the provision of a zeroization function which destroys the keys. Also, upon re-generation of the key, the TOE must ensure that any information about the previous key is wiped from the data structures to ensure that each key is fully unique. The key destruction part of O.HD_DATA is fulfilled by FCS_CKM.4 and FDP_RIP.1a. The data stored on the hard disk of the TOE must be encrypted when forwarded to the hard disk for storage by the host PC and decrypted when returned to the host PC from the hard disk. This is aspect of O.HD_DATA is fulfilled FCS_COP.1b.

Objective	SFRs	Demonstration
O.AUTH	FDP_RIP.1b FIA_AFL.1 FIA_ATD.1 FIA_UAU.1 FIA_UID.1 FPT_FLS.1 FTP_TRP.1 FPT_TST.1b	O.AUTH concerns with the establishment of the authenticity of the users of the TOE and ensuring that only authentic users gain access to the data protected by the TOE. When the TOE is started up during the boot-up of the host PC, an additional authentication dialogue is implemented by the TOE. The integrity of the dialogue must be established prior to the start-up to ensure that the dialogue is in an authentic state and ensures that no unauthorised parties may learn the contents of the authentication exchange. This facet of O.AUTH is fulfilled by FTP_TRP.1. The TOE must also implement a number of self tests on the start-up to ensure authenticity and correct functioning of the authentication functions and authentication data. This feature is fulfilled by addressing FPT_TST.1b. User authentication shall be required prior to any other than the explicitly stated actions are allowed by the TOE. This is fulfilled by FIA_UID.1 and FIA_UAU.1. This feature is also supported by FPT_FLS.1: 1. In case of a power failure as defined in FPT_FLS.1, user authentication must always take place as part of the TOE boot-up sequence. This ensures that tearing attack is foiled and the TOE cannot be powered up so that the previous authentication state could be examined by threat agents and reused to bypass authentication. 2. In case of wake-up from hibernation, as defined in FPT_FLS.1, the user authentication must take place prior to the decryption of the hibernation file to ensure that only legitimate user of the TOE may restore an operational state of the TOE after hibernation. Once the authentication exchange is completed, the TOE shall ensure that the authentication data is wiped off from the memory to ensure that no malicious party monitoring the state of the host PC may learn the authentication data used in the authentication exchange. This is fulfilled by FDP_RIP.1b. Each authenticated user is assigned to a well defined role (FIA_ATD.1) and if the number of failed authentication attempts exceeds a defined threshold, the TOE enters the Lockdown mode. This is fulfilled by FIA_AFL.1.
O.ADMIN	FMT_SMR.1 FMT_SMF.1 FTP_ITC.1	O.ADMIN concerns with the secure administration of the TOE and with ensuring that the TOE recovery feature is only available to legitimate parties. The TOE defines two roles: user and administrator (fulfilled by FMT_SMR.1). The management functions of the TOE are well defined and explicitly enumerated, and are only made available to the users of the TOE assigned to the role of administrator (fulfilled by FMT_SMF.1). In addition to being assigned an administrative role, the user wishing to recover the TOE from Lockdown mode must establish a trusted channel between the administrative console and the TOE. The trusted channel can only be established in the presence of a

Objective	SFRs	Demonstration
		valid response to the recovery challenge. Establishment of the recovery challenge requires participation of both the administrator and the end user to make sure that only upon a valid collaboration can the TOE be recovered into an operational mode from the Lockdown mode. This is fulfilled by FTP_ITC.1.

9.3.2 SAR justification

66 The set of SARs selected for the TOE constitute the entire evaluation assurance level EAL2 with no augmentations. As a basic EAL2 package, the set of SARs is an internally consistent and mutually supportive set of SARs.

67 The TOE is used in a potentially untrusted host PCs but when not in use, in the physical possession of the end user. The relevant attack scenarios are logical attacks occurring through the external interfaces of the TOE by malicious software potentially residing in the host PC.

68 The potentially malicious software running in the host PC can only access the TOE through the authentication interface. Attack scenarios concerning internal interfaces are not accessible as access to those interfaces would require physical probing of the TOE.

69 Consequently, it is sufficient for the TOE to be engineered to demonstrate sufficient assurance against logical attacks by malicious software through externally visible interfaces as demonstrated by EAL2.