Protection Profile for Hardcopy Devices

IPA, NIAP, and the MFP Technical Community

September 10, 2015

Version 1.0

Document History

Version	Date	Comment
1.0	September 10, 2015	Initial release.

Acknowledgements

This protection profile was developed by the Multifunction Printers Technical Community with representatives from industry, U.S. and Japanese Government agencies, Common Criteria Test Laboratories, and international Common Criteria schemes. Information-technology Promotion Agency, Japan and the National Information Assurance Partnership wish to acknowledge and thank the members of this group whose dedicated efforts contributed significantly to the publication.

Table of Contents

List of Tables

1 Protection Profile Introduction (APE_INT, APE_CCL)

1.1 Purpose

¶ 1 The purpose of this Protection Profile (PP) is to facilitate efficient procurement of Commercial Off-The-Shelf (COTS) Hardcopy Devices (HCDs) using the Common Criteria (CC) methodology for information technology security evaluation.

¶ 2 Toward that end, this is a bilateral PP that is based on government procurement requirements from the United States and Japan.

¶ 3 For end customers and general security professionals, this introductory section of this PP uses natural language to describe the primary usage of an HCD, Assumptions about its Operational Environment, security-relevant use cases, and major security functions that support those use cases. The second section introduces the Assets that are protected, Threats that are countered, and the policies that are enforced, by products that conform to this PP. The intent of these sections is to provide sufficient information for potential users to determine if this PP satisfies their security requirements for HCDs.

¶ 4 For HCD developers, CC evaluators, and other CC professionals, this PP also provides standard CC structures and language to define the security problem of the Target of Evaluation (TOE), and to specify the Security Objectives, Security Functional Requirements (SFRs), and Security Assurance Requirements (SARs) that address the security problem. Natural language sections are intended to provide contextual background for standard CC definitions and specifications. The intent of these sections is to provide concise information for developers to implement conforming products for evaluation and for evaluators to test product conformance in an objective and repeatable manner.

1.2 PP Identification and Conformance Claims

¶ 5 **Title:** Protection Profile for Hardcopy Devices

¶ 6 **PP Version:** 1.0 dated September 10, 2015

¶ 7 **Sponsors:** IPA JISEC (Japan), NIAP CCEVS (US)

¶ 8 **Authors:** MFP Technical Community

¶ 9 **Editor:** Brian Smithson, Ricoh Americas

¶ 10 **Keywords:** Multifunction Printer, Multifunction Peripheral, MFP, Multifunction Device, MFD, All-in-one, Hardcopy Device, HCD, Printer, Copier, Photocopier, Scanner, Fax

¶ 11 **CC Conformance:** Common Criteria version: Version 3.1, Release 4, Part 2 (CCMB-2012-09-002) Extended, and Part 3 (CCMB-2012-09-003) Conformant.

¶ 12 **Package conformance**: This Protection Profile does not claim conformance to any packages[1].

¶ 13 **Conformance to other Protection Profiles:** This Protection Profile does not claim conformance to another Protection Profile.

¶ 14 **Conformance to this Protection Profile**: To claim conformance to this Protection Profile, the conforming Security Target must comply with all of the following rules:

¶ 15 1. The TOE must support at least one of the Required Uses scanning, printing, or copying, and must support the Required Uses network communications and administration, described in section 1.3.1.1.

¶ 16 2. Security for all of those Required Uses supported by the TOE must be evaluated, conforming to the requirements of this Protection Profile.

¶ 17 3. If the TOE supports any of the Conditionally Mandatory Uses described in section 1.3.1.2, then that support must be evaluated conforming to the corresponding conditionally mandatory requirements described in Appendix B.

¶ 18 4. The selected communications protocol(s) must be evaluated conforming to the corresponding selection-based protocol requirements in Appendix D.2.

¶ 19 5. The Security Target author may choose to include for evaluation any of the Optional Uses described in section 1.3.1.3. The vendor may choose to evaluate those optional functions as described in Appendix C.

¶ 20 6. The TOE must demonstrate *Exact Conformance*[2]. *Exact Conformance*, as a subset of *Strict Conformance* as defined in Annex D.2 of CC Part 1 (CCMB-

[1] This Protection Profile contains the security assurance requirements required for a Security Target to claim conformance to EAL1 augmented by ASE_SPD.1. The Protection Profile itself conforms to the Standard PP evaluation package as defined in CC Part 3.

[2] Until Exact Conformance is added to the Common Criteria, the requirement for Exact Conformance is a scheme-specific requirement. The CCRA requirement is for Strict Conformance.

2012-09-001), is defined as the ST meeting all of the previous conformance rules. While iteration is allowed, no additional requirements (from the CC parts 2 or 3) are allowed to be included in the ST.

1.3 Overview of the Hardcopy Device

1.3.1 Usage

¶ 21 The Target of Evaluation in this PP is an HCD. HCDs support job functions to convert hardcopy documents into digital form (scanning), convert digital documents into hardcopy form (printing), duplicate hardcopy documents (copying), or transmit documents over a PSTN connection (PSTN faxing). Hardcopy documents typically take the form of paper, but can take other forms (e.g. transparencies).

¶ 22 For the purpose of this PP, a conforming HCD must support at least one of the job functions printing, scanning, or copying and must support the functions network communications and administration (which are described in section 1.3.1.1).

¶ 23 The job functions supported by the HCD and the network communications and administration functions are “Required Uses” of a conforming HCD and are mandatory functions. A conforming HCD may also support “Conditionally Mandatory Uses”. Conditionally Mandatory Uses are optional functions, the presence of which in a HCD is not required for conformance, but which must meet conditionally mandatory requirements if they are present in a HCD.

1.3.1.1 *Required Uses*

¶ 24 The Required Uses that shall be present in a conforming HCD are:

¶ 25 **One or more of the following:**

i. **Printing**: converting an electronic document to hardcopy form, or

ii. **Scanning**: converting a hardcopy document to electronic form, or

iii. **Copying**: duplicating a hardcopy document,

— and —

¶ 26 **Network communications**: sending or receiving documents over a Local Area Network (LAN),

— and —

¶ 27 **Administration**: configuring, auditing, and verifying the security of the HCD.

¶ 28 In other words, a conforming HCD must support at least one of the Required Uses scanning, printing, or copying, and must support the Required Uses network communications and administration.

1.3.1.2 *Conditionally Mandatory Uses*

¶ 29 Conditionally Mandatory Uses that may be present in a conforming HCD are:

¶ 30 **PSTN faxing**: sending and receiving documents over the public switched telephone network (PSTN) using standard facsimile protocols

¶ 31 **Storage and retrieval**: storing electronic documents and retrieving them at a later time

¶ 32 **Field-Replaceable Nonvolatile Storage**: storing documents or confidential system information on Field-Replaceable Nonvolatile Storage Devices.

¶ 33 To conform, the HCD must meet requirements associated with these functions if they are present in the TOE.

1.3.1.3 *Optional Uses*

¶ 34 Optional Uses that may be present in a conforming HCD are:

¶ 35 **Internal Audit Log Storage**: storing audit logs in the HCD

¶ 36 **Image Overwrite:** Actively overwriting residual image data at the conclusion of an image processing job

¶ 37 **Purge Data:** Purging all customer-supplied data from the HCD in preparation for redeployment, decommissioning, or other change in environment.

1.3.2 Boundary of the TOE

¶ 38 The physical boundary of the TOE is the entire HCD product. Options and add-ons that are not security relevant, such as finishers, do not need to be included in the TOE. If it is possible for users to connect personal storage devices (such as portable flash memory devices) to the HCD, those devices and data contained within them are out of scope of the TOE and interfaces to connect such devices should be disabled.

¶ 39 The logical boundary of the TOE includes all security functions related to the Required

Uses of the HCD as described in section 1.3.1.1, all Conditionally Mandatory Uses as described in section 1.3.1.2 that are present in the HCD, and all Optional Uses as described in section 1.3.1.3 that are to be included in the evaluation.

1.3.3 Operational Environment

¶ 40 For the purposes of this PP, HCDs are used in an office environment by commercial, government, or other organizations, and are connected to a wired LAN. If a PSTN fax function is present, then the HCD can also be connected to the PSTN for sending and receiving PSTN faxes.

¶ 41 Users may interact with the HCD through a variety of interfaces:

- A Local User interacts with the HCD using its physical operator console
- A Network User uses interacts with the HCD using programs installed on personal computers or other IT devices external to the HCD which communicate with the HCD through the LAN. This includes the use of general client programs such as web browsers and specific programs such as print or scan drivers.

¶ 42 The HCD and External IT Entities may also interact independently of human User input.

¶ 43 The Operational Environment is assumed to be physically and logically protected from Threats originating from outside of that environment, typically by limiting physical access to the HCD and connecting it to a LAN that is protected from the public Internet.

1.4 Security Use Cases of the HCD

¶ 44 Security use cases illustrate a User's security expectations as they use the HCD.

1.4.1 Required Use Cases

¶ 45 The security-relevant use cases for Required Uses of a conforming HCD are:

¶ 46 **1. One or more of the following:**

a) **Printing**: A Network User sends a Document from an External IT Entity to the HCD over a LAN with instructions for printing. The HCD has the capability to protect the User's Document from unauthorized disclosure or alteration while it is in transit to the HCD, in Temporary Storage in the HCD, and before printed output is released to a User.

b) **Scanning**: A Local User initiates scanning a Document on the HCD and the HCD sends the digital image to an External IT Entity. The HCD has the capability to protect the User's Document from unauthorized disclosure or alteration while it is in Temporary Storage in the HCD and while it is in transit to the External IT Entity.

c) **Copying**: A Local User scans a Document on the HCD and the HCD prints the Document. The HCD has the capability to protect the User's Document from unauthorized disclosure and alteration while it is in Temporary Storage in the HCD.

¶ 47 **2. Configuration**: A Local or Network User with administrative privileges configures the security settings of the HCD. The HCD has the capability to assign Users to roles that distinguish Users who can perform administrative functions from Users who can perform User functions. The HCD also has the capability to protect its security settings from unauthorized disclosure and alteration when they are stored in the HCD and in transit to or from an External IT Entity.

¶ 48 **3. Auditing**: Authorized personnel monitor security-relevant events in an audit log. The HCD generates audit log records when security-relevant events occur. It is mandatory that the HCD is able to securely transmit audit logs to an External IT Entity for storage, and the HCD has the capability to protect it from unauthorized disclosure or alteration while in transit to the External IT Entity.

¶ 49 **4. Verifying software updates**: Authorized personnel install updated software on the HCD. The HCD ensures that only authorized personnel are permitted to install software, has the capability to help the installer to verify the authenticity of the software update.

¶ 50 **5. Verifying HCD function**: The HCD checks itself for malfunctions by performing a self-test each time that it is powered on.

1.4.2 Conditionally Mandatory Use Cases

¶ 51 Security-relevant use cases for Conditionally Mandatory Uses (if present) of a conforming HCD may include:

¶ 52 **Sending PSTN faxes**: A Local User scans a Document on the HCD, or a Network User sends a Document from an External IT Entity to the HCD; the User provides instructions for sending it to a remote PSTN fax destination; the HCD sends a

facsimile of the Document over the PSTN to the PSTN fax destination using standard PSTN fax protocols. The HCD has the capability to protect the Network User's Document from unauthorized disclosure and alteration while in transit on the LAN. The HCD also has the capability to protect the User's Document from unauthorized disclosure and alteration while in Temporary Storage in the HCD.

¶ 53 **Receiving PSTN faxes**: A remote PSTN fax sender sends a facsimile of a Document over the PSTN to the HCD using standard PSTN fax protocols. The HCD has the capability to protect received PSTN faxes from unauthorized disclosure and alteration while it is present in the HCD. Further, the HCD has the capability to ensure that the PSTN fax modem is not used to access the LAN.

¶ 54 **Storing and retrieving Documents**: A Local or Network User instructs the HCD to store or retrieve an electronic Document in the HCD. The sources and destinations of such Documents may be any of the other operations such as scanning, printing, or PSTN faxing. The HCD has the capability to protect such Documents from unauthorized disclosure and alteration while in transit and in storage in the HCD.

¶ 55 **Field-Replaceable Nonvolatile Storage Devices**: Authorized personnel remove the HCD from service in its Operational Environment to perform preventative maintenance, repairs, or other servicing-related operations. The HCD has the capability to protect documents or confidential system information that may be present in Field-Replaceable Nonvolatile Storage Devices from exposure if such a device is removed from the HCD.

1.4.3 Optional Use Cases

¶ 56 Security-relevant use cases for Optional Uses (if present) of a conforming HCD may include:

¶ 57 **Internal Audit Log Storage**: If the audit log can also be stored in the HCD, the HCD has the capability to protect its audit log from unauthorized disclosure and alteration.

¶ 58 **Image Overwrite**: At the conclusion of an image processing job, residual image data may be present in the HCD. The HCD has the capability to actively overwrite such image data.

¶ 59 **Redeploying or Decommissioning the HCD**: Authorized personnel remove the

HCD from service in its Operational Environment to move it to a different Operational Environment, to permanently remove it from operation, or otherwise change its ownership. The HCD has the capability to make all customer data that may be present in the HCD unavailable for recovery if it is removed from the Operational Environment.

1.5 Major Security Functions of the HCD

¶ 60 To support the use cases in section 1.4, a conforming HCD provides the following security functions:

1. Identification, authentication, and authorization to use HCD functions
2. Access control
3. Encryption
4. Trusted communications
5. Administrative roles
6. Auditing
7. Trusted operation
8. PSTN fax-network separation (if PSTN fax function is present)
9. Data clearing and purging (optional)

¶ 61 Each of these functions is described in the next subsections.

1.5.1 Identification, Authentication, and Authorization

¶ 62 User identification, authentication, and authorization ensure that functions of the HCD are accessible only to Users who have been authorized by an Administrator. User identification and authentication is also used as the basis for access control and administrative roles and helps associate security-relevant events and HCD use with specific Users. Identification and authentication may be performed by the HCD or by an external server.

1.5.2 Access Control

¶ 63 Access controls ensure that Documents, information related to Document Processing, and

security-relevant data are accessible only to Users who have appropriate access permissions.

1.5.3 Data Encryption

¶ 64 Data encryption ensures that data assets cannot be accessed while in transit on the LAN.

¶ 65 By policy, data encryption is also used to protect documents and confidential system information on Field-Replaceable Nonvolatile Storage Devices to protect such data if such a device is removed from the HCD.

¶ 66 The effectiveness of data encryption is assured through the use of internationally accepted cryptographic algorithms.

1.5.4 Trusted Communications

¶ 67 Trusted communication paths are established to ensure that communications with the HCD are performed with known endpoints.

1.5.5 Administrative Roles

¶ 68 Role-based access controls ensure that the ability to configure the security settings of the HCD is available only to Users who have been authorized with an Administrator role.

1.5.6 Auditing

¶ 69 Audit logs are generated by the HCD to ensure that security-relevant events and HCD use can be monitored by authorized personnel. The HCD must generate audit logs and securely transmit them to an External IT entity for storage. Optionally, audit logs may also be stored in the HCD where they can be reviewed by an Administrator.

1.5.7 Trusted Operation

¶ 70 Software updates to the HCD are verified to ensure the authenticity of the software before applying the update. The HCD performs self-tests to ensure that its operation is not disrupted by some detectable malfunctions.

1.5.8 PSTN Fax-Network Separation

¶ 71 If a conforming HCD has a PSTN fax function, PSTN fax-network separation ensures that the PSTN fax modem is not used to create a data bridge between the PSTN and the LAN.

1.5.9 **Data Clearing and Purging**

¶ 72 Optionally, an HCD may provide functions that actively overwrite image data, or that purge all customer-supplied information at the request of an authorized Administrator. These are discussed in Appendix C.

2 Security Problem Definition (APE_SPD)

¶ 73 The Security Problem Definition (SPD) is divided into two parts. This first part describes Assets, Threats, and Organizational Security Policies, in narrative form. [Brackets] indicate a reference to the second part, formal definitions of Users, Assets, Threats, Organizational Security Policies, and Assumptions, which appear in Appendix A.

¶ 74 Note: From this point in the document, the Target of Evaluation will be referred to by the acronym “TOE” (Target of Evaluation) instead of by the product category “HCD” (Hardcopy Device).

2.1 Users

¶ 75 A conforming TOE must define at least the following two User roles:

1. Normal Users [U.NORMAL] who are identified and authenticated and do not have an administrative role.
2. Administrators [U.ADMIN] who are identified and authenticated and have an administrative role.

¶ 76 A conforming TOE may allow additional roles, sub-roles, or groups. In particular, a conforming TOE may allow several administrative roles that have authority to administer different aspects of the TOE.

¶ 77 Note that a User can be a human user or an external IT entity.

¶ 78 Additional details about Users are in Appendix A.1.

2.2 Assets

¶ 79 From a User’s perspective, the primary Asset to be protected in a TOE is User Document Data [D.USER.DOC]. A User’s job instructions, User Job Data [D.USER.JOB] (information related to a User’s Document or Document Processing Job), may also be protected if their compromise impacts the protection of User Document Data. Together, User Document Data and User Job Data are considered to be User Data.

¶ 80 As an illustrative example, data sent by a Network User for printing contains a User’s Document [D.USER.DOC] which must not be accessed by anyone else, and job instructions such as the destination to send scanned Documents [D.USER.JOB] which must not be altered by anyone else.

¶ 81 From an Administrator's perspective, the primary Asset to be protected in a TOE is data that is used to configure and monitor the secure operation of the TOE. This kind of data is considered to be TOE Security Functionality (TSF) Data.

¶ 82 There are two broad categories for this kind of data:

1. Protected TSF Data, which may be read by any User but must be protected from unauthorized modification and deletion [D.TSF.PROT]; and,
2. Confidential TSF Data, which may neither be read nor modified or deleted except by authorized Users [D.TSF.CONF].

¶ 83 An illustrative example is data that is used by the TOE to identify and authenticate authorized Users. Typically, a username that is used for identification may be read by anyone but must be protected from unauthorized modification and deletion [D.TSF.PROT]. In contrast, a User's password that is used for authentication must be confidential, prohibiting any Unauthorized Access [D.TSF.CONF].

¶ 84 If TSF Data is compromised, it can be used for a variety of malicious purposes that include elevation of privileges, accessing stored Documents, redirecting the destination of processed Documents, masquerading as an authorized User or Administrator, altering the operating software of the TOE, and attacking External IT Entities.

¶ 85 In a conforming TOE, TSF Data is clearly identified and categorized as either Protected TSF Data or Confidential TSF Data.

¶ 86 From a network security perspective, it is important to ensure the secure operation of the TOE and other IT entities in its Operational Environment. Since the Operational Environment is outside of the TOE, Organizational Security Policies are employed to address protection of the Operational Environment.

¶ 87 Additional details about assets are in Appendix A.2.

2.3 Threats

¶ 88 The following are Threats against the TOE that are countered by conforming products. Additional details about threats are in Appendix A.3.

2.3.1 Unauthorized Access to User Data

¶ 89 An attacker may access (read, modify, or delete) User Document Data or change (modify

or delete) User Job Data in the TOE through one of the TOE's interfaces [T.UNAUTHORIZED_ACCESS]. For example, depending on the design of the TOE, the attacker might access the printed output of a Network User's print job, or modify the instructions for a job that is waiting in a queue, or read User Document Data that is in a User's private or group storage area.

2.3.2 Unauthorized Access to TSF Data

¶ 90 An attacker may gain Unauthorized Access to TSF Data in the TOE through one of the TOE's interfaces [T.TSF_COMPROMISE]. For example, depending on the design of the TOE, the attacker might use Unauthorized Access to TSF Data to elevate their own privileges, alter an Address Book to redirect output to a different destination, or use the TOE's Credentials to gain access to an external server.

¶ 91 An attacker may cause the installation of unauthorized software on the TOE [T.UNAUTHORIZED_UPDATE]. For example, unauthorized software could be used to gain access to information that is processed by the TOE, or to attack other systems on the LAN.

2.3.3 Network Communication Attacks

¶ 92 An attacker may access data in transit or otherwise compromise the security of the TOE by monitoring or manipulating network communication [T.NET_COMRPOMISE]. For example, here are several ways that network communications could be compromised: By monitoring clear-text communications on a wired LAN, the attacker might obtain User Document Data, User Credentials, or system Credentials, or hijack an interactive session. The attacker might record and replay a network communication session in order to log into the TOE as an authorized User to access Documents or as an authorized Administrator to change security settings. The attacker might masquerade as a trusted system on the LAN in order to receive outgoing scan jobs, to record the transmission of system Credentials, or to send malicious data to the TOE.

2.3.4 Malfunction

¶ 93 A malfunction of the TSF may cause loss of security if the TOE is permitted to operate while in a degraded state [T.TSF_FAILURE]. Hardware or software malfunctions can produce unpredictable results, with a possibility that security functions will not operate correctly.

2.4 Organizational Security Policies

¶ 94 The following are Organizational Security Policies[3] (OSPs) that are upheld by conforming products. Additional details about OSPs are in Appendix A.4.

2.4.1 User Authorization

¶ 95 Users must be authorized before performing Document Processing and administrative functions [P.AUTHORIZATION]. Authorization allows the TOE Owner to control who is able to use the resources of the TOE and who is permitted to perform administrative functions.

2.4.2 Auditing

¶ 96 Security-relevant activities must be audited and the log of such actions must be protected and transmitted to an External IT Entity [P.AUDIT]. Stored on an External IT Entity (or, optionally, also in the TOE), an audit trail makes it possible for authorized personnel to review and identify suspicious activities and to account for TOE use as may be required by site policy or regulations.

2.4.3 Protected Communications

¶ 97 The TOE must be able to identify itself to other devices on the LAN [P.COMMS_PROTECTION]. Assuring identification helps prevent an attacker from masquerading as the TOE in order to receive incoming print jobs, recording the transmission of User Credentials, or sending malicious data to External IT Entities.

2.4.4 Storage Encryption (conditionally mandatory)

¶ 98 If the TOE stores User Document Data or Confidential TSF Data on Field-Replaceable Nonvolatile Storage Devices[4], it will encrypt such data on those devices [P.STORAGE_ENCRYPTION]. Data is assumed to be protected by the TSF when the TOE is operating in its Operational Environment. However, if Field-Replaceable

[3] Organizational Security Policy is a term that encompasses the security policy or policies that are supported / enforced by the TOE. That is, the TOE supports the requirements of organizations that need to enforce the identified policies. Site policy is a general term referring to security policies of customers which cannot be specified in a PP.

[4] A "Field-Replaceable Nonvolatile Storage Device" is any Field-Replaceable Unit (FRU) for which the primary purpose is to provide nonvolatile storage. This OSP does not apply to storage devices that are a non-field-replaceable component of a larger FRU that is not primarily used for storage.

Nonvolatile Storage Devices are removed from the TOE for Servicing, redeployment to another environment, or decommissioning, an attacker may be able to expose or modify User Document Data or Confidential TSF Data. Encrypting such data prevents the attacker from doing so without access to encryption keys or keying material.

¶ 99 Cleartext keys, submasks, random numbers, or any other values that contribute to the creation of encryption keys for Field-Replaceable Nonvolatile Storage of User Document Data or Confidential TSF Data must be protected from unauthorized access and must not be stored on that storage device [P.KEY_MATERIAL]. Unauthorized possession of key material in cleartext may allow an attacker to decrypt User Document Data or Confidential TSF Data.

2.4.5 PSTN Fax-Network Separation (conditionally mandatory)

¶ 100 If the TOE includes a PSTN fax function, it will ensure separation between the PSTN fax line and the LAN [P.FAX_FLOW]. The TOE is assumed to be in an Operational Environment that is protected, such as by an external firewall. However, the PSTN fax modem may be connected to a public switched telephone network. Ensuring separation of the PSTN fax and network prevents an attacker from using the PSTN fax modem to bypass the firewall or other external protection to access the protected environment.

2.4.6 Image Overwrite (optional)

¶ 101 Upon completion or cancellation of a Document Processing job, the TOE shall overwrite residual image data from its Field-Replaceable Nonvolatile Storage Devices [P.IMAGE_OVERWRITE]. A customer may be concerned that image data that has been dereferenced by the TOE operating software may remain on Field-Replaceable Nonvolatile Storage Devices in the TOE after a Document Processing job has been completed or cancelled. Such customers desire that the image data be made unavailable by overwriting it with other data.

2.4.7 Purge Data (optional)

¶ 102 The TOE shall provide a function that an authorized administrator can invoke to make all customer-supplied User Data and TSF Data permanently irretrievable from Nonvolatile Storage Devices [P.PURGE_DATA]. A customer may be concerned that data which is considered confidential in the Operational Environment may remain in Nonvolatile Storage Devices in the TOE after the TOE is permanently removed from its Operational Environment to be decommissioned from service or to be redeployed to a different

Operational Environment. Such customers desire that all customer-supplied User Data and TSF Data be purged from the TOE so that it cannot be retrieved outside of the Operational Environment.

2.5 Assumptions

¶ 103 The following assumptions must be upheld so that the objectives and requirements can effectively counter the threats described in this Protection Profile. Additional details about assumptions are in Appendix A.5.

2.5.1 Physical Security

¶ 104 Physical security, commensurate with the value of the TOE and the data it stores or processes, is assumed to be provided by the environment [A.PHYSICAL]. The TOE is assumed to be located in a physical environment that is controlled or monitored such that a physical attack is prevented or detected.

2.5.2 Network Security

¶ 105 The Operational Environment is assumed to protect the TOE from direct, public access to its LAN interface [A.NETWORK]. The TOE is not intended to withstand network-based attacks from an unmanaged network environment.

2.5.3 Administrator Trust

¶ 106 TOE Administrators are trusted to administer the TOE according to site security policies [A.TRUSTED_ADMIN]. It is the responsibility of the TOE Owner to only authorize administrators who are trusted to configure and operate the TOE according to site policies and to not use their privileges for malicious purposes.

2.5.4 User Training

¶ 107 Authorized Users are trained to use the TOE according to site security policies [A.TRAINED_USERS]. It is the responsibility of the TOE Owner to only authorize Users who are trained to use the TOE according to site policies.

3 Security Objectives (APE_OBJ)

3.1 Security Objectives for the TOE

¶ 108 The following Security Objectives must be fulfilled by the TOE. Additional details about objectives for the TOE are in Appendices A.6 and A.7.

3.1.1 User Authorization

¶ 109 The TOE shall perform authorization of Users in accordance with security policies [O.USER_AUTHORIZATION].

¶ 110 This objective supports the policy that Users are authorized to administer the TOE or perform Document Processing functions that consume TOE resources. Users must be authorized to perform any of the Document Processing functions present in the TOE.

¶ 111 The mechanism for authorization is implemented within the TOE, and it may also depend on a trusted External IT Entity. If a conforming TOE supports more than one mechanism, then each should be evaluated as separate modes of operation.

¶ 112 In the case of printing (if that function is present in the TOE), User authorization may take place after the job has been submitted but must take place before printed output is made available to the User.

¶ 113 Users must be authorized to perform PSTN fax sending functions and document storage and retrieval functions, if such functions are provided by the conforming TOE.

¶ 114 Note that the TOE can receive a PSTN fax without any User authorization, but the received Document is subject to access controls.

3.1.2 User Identification and Authentication

¶ 115 The TOE shall perform identification and authentication of Users for operations that require access control, User authorization, or Administrator roles [O.USER_I&A].

¶ 116 The mechanism for identification and authentication (I&A) is implemented within the TOE, and it may also depend on a trusted External IT Entity (e.g., LDAP, Kerberos, or Active Directory). If a conforming TOE supports more than one mechanism, then each should be evaluated as separate modes of operation.

3.1.3 Access Control

¶ 117 The TOE shall enforce access controls to protect User Data and TSF Data in accordance with security policies [O.ACCESS_CONTROL].

¶ 118 The guiding principles for access control security policies in this PP are:

- User Document Data [D.USER.DOC] can be accessed only by the Document owner or an Administrator.
- User Job Data [D.USER.JOB] can be read by any User but can be modified only by the Job Owner or an Administrator.
- Protected TSF Data [D.TSF.PROT] are data that can be read by any User but can be modified only by an Administrator or (in certain cases) a Normal User who is the owner of or otherwise associated with that data.
- Confidential TSF Data [D.TSF.CONF] are data that can only be accessed by an Administrator or (in certain cases) a Normal User who is the owner of or otherwise associated with that data.

¶ 119 The Security Target of a conforming TOE must clearly specify its access control policies for User Data and TSF Data.

3.1.4 Administrator Roles

¶ 120 The TOE shall ensure that only authorized Administrators are permitted to perform administrator functions [O.ADMIN_ROLES].

¶ 121 This objective addresses the need to have at least one Administrator role that is distinct from Normal Users. A conforming TOE may have specialized Administrator sub-roles, such as for device management, network management, or audit management.

3.1.5 Software Update Verification

¶ 122 The TOE shall provide mechanisms to verify the authenticity of software updates [O.UPDATE_VERIFICATION].

¶ 123 This objective addresses the concern that malicious software may be introduced into the TOE as a software update. Verifying authenticity, such as with a digital signature or published hash, is required. Access control by itself does not satisfy this objective.

3.1.6 Self-test

¶ 124 The TOE shall test some subset of its security functionality to help ensure that subset is operating properly [O.TSF_SELF_TEST].

¶ 125 A malfunction of the TOE may compromise its security if the malfunction is not detected and the TOE is allowed to operate. Self-test is intended to detect such malfunctions. It is performed during power-up.

3.1.7 Communications Protection

¶ 126 The TOE shall have the capability to protect LAN communications of User Data and TSF Data from Unauthorized Access, replay, and source/destination spoofing [O.COMMS_PROTECTION].

¶ 127 This objective addresses the common concerns of network communications:

- Sensitive data or Credentials are obtained by monitoring LAN data outside of the TOE.
- A successfully authenticated session is captured and replayed on the LAN, permitting the attacker to masquerade as the authenticated User.
- Sensitive data or Credentials are obtained by redirecting communications from the TOE or from an External IT Entity to a malevolent destination.

3.1.8 Auditing

¶ 128 The TOE shall generate audit data, and be capable of sending it to a trusted External IT Entity. Optionally, it may store audit data in the TOE [O.AUDIT].

¶ 129 The TOE must be able to send audit data to a trusted External IT Entity (e.g., an audit server such as a syslog server). Audit data may also be stored in the TOE with appropriate access controls to ensure confidentiality and integrity. If a conforming TOE supports both mechanisms, then each should be evaluated as separate modes of operation.

3.1.9 Storage Encryption (conditionally mandatory)

¶ 130 If the TOE stores User Document Data or Confidential TSF Data in Field-Replaceable Nonvolatile Storage devices, then the TOE shall encrypt such data on those devices. [O.STORAGE_ENCRYPTION].

¶ 131 This objective addresses the concern that User Document Data or Confidential TSF Data

on a Field-Replaceable Nonvolatile Storage Device may be exposed if the device is removed from the TOE, such as for Servicing, Redeployment to another environment, or Decommissioning.

3.1.10 Protection of Key Material (conditionally mandatory)

¶ 132 The TOE shall protect from unauthorized access any cleartext keys, submasks, random numbers, or other values that contribute to the creation of encryption keys for storage of User Document Data or Confidential TSF Data in Field-Replaceable Nonvolatile Storage Devices; The TOE shall ensure that such key material is not stored in cleartext on the storage device that uses that material [O.KEY_MATERIAL].

¶ 133 This objective addresses the concern that unauthorized possession of keys or key material may be used to decrypt User Document Data or Confidential TSF Data.

3.1.11 PSTN Fax-Network Separation (conditionally mandatory)

¶ 134 If the TOE provides a PSTN fax function, then the TOE shall ensure separation of the PSTN fax telephone line and the LAN, by system design or active security function [O.FAX_NET_SEPARATION].

¶ 135 This objective addresses customer concerns about having a telephone line connected to a device that is inside their firewall. Depending on implementation, it may be satisfied in different ways, such as by system architecture (no data path from the PSTN fax interface to the network interface), by system design (fax chipset recognizes only PSTN fax protocols), or by active security function (flow control).

3.1.12 Image Overwrite (optional)

¶ 136 Upon completion or cancellation of a Document Processing job, the TOE shall overwrite residual image data in its Field-Replaceable Nonvolatile Storage Devices [O.IMAGE_OVERWRITE]. This objective addresses customer concerns that image data may remain on Field-Replaceable Nonvolatile Storage Devices in the TOE after a Document Processing job has been completed or cancelled.

3.1.13 Purge Data (optional)

¶ 137 The TOE provides a function that an authorized administrator can invoke to make all customer-supplied User Data and TSF Data permanently irretrievable from Nonvolatile Storage Devices [O.PURGE_DATA]. This objective addresses customer concerns that

data that is protected in the Operational Environment may remain in Nonvolatile Storage Devices after the TOE is permanently removed from its Operational Environment to be decommissioned from service or to be redeployed to a different Operational Environment.

3.2 Security Objectives for the Operational Environment

¶ 138 The following Security Objectives must be provided by the Operational Environment. Additional details about objectives for the Operational Environment are in Appendix A.7.

3.2.1 Physical Protection

¶ 139 The Operational Environment shall provide physical security, commensurate with the value of the TOE and the data it stores or processes [OE.PHYSICAL_PROTECTION].

¶ 140 Due to its intended function, this kind of TOE must be physically accessible to authorized Users, but it is not expected to be hardened against physical attacks. Therefore, the environment must provide an appropriate level of physical protection or monitoring to prevent physical attacks.

3.2.2 Network Protection

¶ 141 The Operational Environment shall provide network security to protect the TOE from direct, public access to its LAN interface [OE.NETWORK_PROTECTION].

¶ 142 This kind of TOE is not intended to be directly connected to a hostile network. Therefore, the environment must provide an appropriate level of network isolation.

3.2.3 Trusted Administrators

¶ 143 The TOE Owner shall establish trust that Administrators will not use their privileges for malicious purposes [OE.ADMIN_TRUST].

¶ 144 Administrators have privileges that can be misused for malicious purposes. It is the responsibility of the TOE Owner to grant administrator privileges only to individuals whom the TOE Owner trusts.

3.2.4 Trained Users

¶ 145 The TOE Owner shall ensure that Users are aware of site security policies and have the competence to follow them [OE.USER_TRAINING].

¶ 146 Site security depends on a combination of TOE security functions and appropriate use of

those functions by Normal Users. Manufacturers may provide guidance to the TOE Owner regarding the TOE security functions that apply to Normal Users.

3.2.5 Trained Administrators

¶ 147 The TOE Owner shall ensure that Administrators are aware of site security policies and have the competence to use manufacturer's guidance to correctly configure the TOE and protect passwords and keys accordingly [OE.ADMIN_TRAINING].

¶ 148 This kind of TOE may have many options for enabling and disabling security functions. Administrators must be able to understand and configure the TOE security functions to enforce site security policies.

4 Security Functional Requirements (APE_REQ, APE_ECD)

4.1 Notational Conventions

¶ 149 **Bold** typeface indicates the portion of an SFR that has been completed or refined in this Protection Profile, relative to the original SFR definition in Common Criteria Part 2 or to its Extended Component Definition.

¶ 150 *Italic* typeface indicates the text within an SFR that must be selected and/or completed by the ST Author in a conforming Security Target.

¶ 151 ***Bold italic*** typeface indicates the portion of an SFR that has been partially completed or refined in this Protection Profile, relative to the original SFR definition in Common Criteria Part 2 or to its Extended Component Definition. These also must be selected and/or completed by the ST Author in a conforming Security Target.

¶ 152 SFR components that are followed by a letter in parentheses, e.g., (a), (b)… represent required iterations.

¶ 153 Extended components are identified by "_EXT" appended to the SFR identifier.

4.2 Extended Components

¶ 154 Extended component definitions are listed in Appendix A.9.

4.3 Class FAU: Security Audit

4.3.1 FAU_GEN.1 Audit data generation

(for O.AUDIT)

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

¶ 155 **FAU_GEN.1.1** The TSF shall be able to generate an audit record of the following auditable events:

¶ 156 a) Start-up and shutdown of the audit functions;

¶ 157 b) All auditable events for the **not specified** level of audit; and

¶ 158 c) All auditable events specified in Table 1, [assignment: *other specifically*

defined auditable events].

¶ 159 **FAU_GEN.1.2** The TSF shall record within each audit record at least the following information:

¶ 160 a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

¶ 161 b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **additional information specified in Table 1**, [assignment: *other audit relevant information*].

Table 1 Auditable Events

Auditable event	Relevant SFR	Additional information
Job completion	FDP_ACF.1	Type of job
Unsuccessful User authentication	FIA_UAU.1	None
Unsuccessful User identification	FIA_UID.1	None
Use of management functions	FMT_SMF.1	None
Modification to the group of Users that are part of a role	FMT_SMR.1	None
Changes to the time	FPT_STM.1	None
Failure to establish session	FTP_ITC.1, FTP_TRP.1(a), FTP_TRP.1(b)	Reason for failure

¶ 162 ***Application Note:***

¶ 163 *In cases where user identification events are inseparable from user authentication events, they may be considered to be a single event for audit purposes.*

¶ 164 *Regarding FMT_SMR.1, if the relationship between users and roles is not modifiable, its auditable event cannot be generated and the requirement to generate an audit record can be ignored.*

¶ 165 *The ST author can include other auditable events directly in the table; they are not limited to the list presented.*

¶ 166 **Assurance Activity:**

¶ 167 ***TSS:***

¶ 168 The evaluator shall check the TOE Summary Specification (TSS) to ensure that auditable events and its recorded information are consistent with the definition of the SFR.

¶ 169 ***Operational Guidance:***

¶ 170 The evaluator shall check the guidance documents to ensure that auditable events and its recorded information are consistent with the definition of the SFRs.

¶ 171 ***Test:***

¶ 172 The evaluator shall also perform the following tests:

¶ 173 The evaluator shall check to ensure that the audit record of each of the auditable events described in Table 1 is appropriately generated.

¶ 174 The evaluator shall check a representative sample of methods for generating auditable events, if there are multiple methods.

¶ 175 The evaluator shall check that FIA_UAU.1 events have been generated for each mechanism, if there are several different I&A mechanisms.

4.3.2 FAU_GEN.2 User identity association

(for O.AUDIT)

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

FIA_UID.1 Timing of identification

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

¶ 177 **Assurance Activity:**

¶ 178 The Assurance Activities for FAU_GEN.1 address this SFR.

4.3.3 FAU_STG_EXT.1 Extended: External Audit Trail Storage

(for O.AUDIT)

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation,

FTP_ITC.1 Inter-TSF trusted channel.

¶ 179 **FAU_STG_EXT.1.1** The TSF shall be able to transmit the generated audit data to an External IT Entity using a trusted channel according to FTP_ITC.1.

¶ 180 **Assurance Activity:**

¶ 181 ***TSS:***

¶ 182 The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism.

¶ 183 The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and how these records are protected against unauthorized access. The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server. For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store

used as a buffer and "cleared" periodically by sending the data to the audit server.

¶ 184 ***Operational Guidance:***

¶ 185 The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server. The evaluator shall perform the following test for this requirement:

¶ 186 ***Test:***

¶ 187 Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.

4.4 Class FCO: Communication

¶ 188 There are no class FCO requirements.

4.5 Class FCS: Cryptographic Support

4.5.1 FCS_CKM.1(a) Cryptographic Key Generation (for asymmetric keys)

(for O.COMMS_PROTECTION)

Hierarchical to: No other components.

Dependencies: [~~FCS_CKM.2 Cryptographic key distribution, or~~

FCS_COP.1(b) Cryptographic Operation (for signature generation/ verification)]

FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

¶ 189 **FCS_CKM.1.1(a) Refinement**: The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance **with [selection:**

- ***NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;***
- ***NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [selection: P-521, no other curves] (as defined in FIPS PUB 186-4, "Digital Signature Standard")***
- ***NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes***

¶ 190 **] and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.**

¶ 191 ***Application Note:***

¶ 192 *The ST author selects the key generation scheme used for key establishment and device authentication. If multiple schemes are supported, then the ST author should iterate this component to capture this capability. When key generation is used for device authentication, the public key is expected to be associated with an X.509v3 certificate. If the TOE acts as a receiver in the RSA key establishment scheme, the TOE does not need to implement RSA key generation.*

¶ 193 *Since the domain parameters to be used are specified by the requirements of the protocol in this PP, it is not expected that the TOE will generate domain parameters, and therefore there is no additional domain parameter validation needed when the TOE complies with the protocols specified in this PP.*

¶ 194 *SP 800-56B references (but does not mandate) key generation according to FIPS 186-3. For purposes of compliance in this version of the HCD PP, RSA key pair generation according to FIPS 186-4 is allowed in order for the TOE to claim conformance to SP 800-56B.*

¶ 195 *The generated key strength of 2048-bit DSA and rDSA keys need to be equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.*

¶ 196 **Assurance Activity:**

¶ 197 ***TSS:***

¶ 198 The evaluator shall ensure that the TSS contains a description of how the TSF complies with 800-56A and/or 800-56B, depending on the selections made. This description shall indicate the sections in 800-56A and/or 800-56B that are implemented by the TSF, and the evaluator shall ensure that key establishment is among those sections that the TSF claims to implement.

¶ 199 Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described in the TSS.

¶ 200 The TSS may refer to the Key Management Description (KMD), described in Appendix F , that may not be made available to the public.

¶ 201 ***Test:***

¶ 202 The evaluator shall use the key pair generation portions of "The FIPS 186-4 Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-4 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and "The 186-4 RSA Validation System (RSA2VS)" as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test.

4.5.2 FCS_CKM.1(b) Cryptographic key generation (Symmetric Keys)

(for O.COMMS_PROTECTION, O.STORAGE_ENCRYPTION)

Hierarchical to: No other components.

Dependencies: [~~FCS_CKM.2 Cryptographic key distribution, or~~
FCS_COP.1(f) Cryptographic operation (Key Encryption)]
FCS_CKM_EXT.4 Extended: Cryptographic Key Material

Destruction

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

¶ 203 **FCS_CKM.1.1(b) Refinement:** The TSF shall generate **symmetric** 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.**

¶ 204 ***Application Note:***

¶ 205 *Symmetric keys may be used to generate keys along the key chain.*

¶ 206 **Assurance activity:**

¶ 207 ***TSS:***

¶ 208 The evaluator shall review the TSS to determine that it describes how the functionality described by FCS_RBG_EXT.1 is invoked.

¶ 209 ***KMD:***

¶ 210 If the TOE is relying on random number generation from a third-party source, the KMD needs to describe the function call and parameters used when calling the third-party DRBG function. Also, the KMD needs to include a short description of the vendor's assumption for the amount of entropy seeding the third-party DRBG. The evaluator uses the description of the RBG functionality in FCS_RBG_EXT or the KMD 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(d)).

¶ 211 The KMD is described in Appendix F.

4.5.3 FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

(for O.COMMS_PROTECTION, O.STORAGE_ENCRYPTION, O.PURGE_DATA)

Hierarchical to: No other components.

Dependencies: [FCS_CKM.1(a) Cryptographic Key Generation (for asymmetric keys), or

FCS_CKM.1(b) Cryptographic key generation (Symmetric Keys)],

FCS_CKM.4 Cryptographic key destruction

¶ 212 **FCS_CKM_EXT.4.1** The TSF shall destroy **all plaintext secret and private cryptographic keys and cryptographic critical security parameters** when no longer needed.

¶ 213 ***Application Note:***

¶ 214 *"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".*

¶ 215 *Keys, including intermediate keys and key material that are no longer needed are destroyed by using an approved method, FCS_CKM.4.1. Examples of keys are intermediate keys, submasks, and BEV. There may be instances where keys or key material that are contained in persistent storage are no longer needed and require destruction. Based on their implementation, vendors will explain when certain keys are no longer needed. There are multiple situations in which key material is no longer necessary, for example, a wrapped key may need to be destroyed when a password is changed. However, there are instances when keys are allowed to remain in memory, for example, a device identification key.*

¶ 216 **Assurance activity:**

¶ 217 ***TSS:***

¶ 218 The evaluator shall verify the TSS provides a high level description of what it means for keys and key material to be no longer needed and when then should be expected to be destroyed.

¶ 219 ***KMD:***

¶ 220 The evaluator shall verify the Key Management Description (KMD) includes a description of the areas where keys and key material reside and when the keys and key material are no longer needed.

¶ 221 The evaluator shall verify the KMD includes a key lifecycle, that includes a description where key material reside, how the key material is used, how it is determined that keys and key material are no longer needed, and how the material is destroyed once it is not needed and that the documentation in the

KMD follows FCS_CKM.4 for the destruction.

4.5.4 FCS_CKM.4 Cryptographic key destruction

(for O.COMMS_PROTECTION, O.STORAGE_ENCRYPTION, O.PURGE_DATA)

Hierarchical to: No other components.

Dependencies: [FCS_CKM.1(a) Cryptographic Key Generation (for asymmetric keys), or

FCS_CKM.1(b) Cryptographic key generation (Symmetric Keys)]

¶ 222 **FCS_CKM.4.1 Refinement:** The TSF shall **destroy** cryptographic keys in accordance with a specified cryptographic key **destruction** method [selection:

¶ 223 **For volatile memory, the destruction shall be executed by [selection: *powering off a device,* [assignment: *other mechanism that ensures keys are destroyed*]].**

¶ 224 **For nonvolatile storage, the destruction shall be executed by a [selection: *single, three or more times*] overwrite of key data storage location consisting of [selection: *a pseudo random pattern using the TSF's RBG (as specified in FCS_RBG_EXT.1), a static pattern*], followed by a [selection: *read-verify, none*]. If read-verification of the overwritten data fails, the process shall be repeated again;**

¶ 225] that meets the following: [**selection: *NIST SP800-88, no standard***].

¶ 226 ***Application Note:***

¶ 227 *Keys, including intermediate keys and key material that are no longer needed are destroyed in volatile memory by using one of these approved methods. In these cases, the destruction method conforms to one of methods specified in this requirement. This requirement calls out the method for performing Cryptographic Erase and is considered a well-defined term for the destruction of key information. Some solutions support write access to media locations where keys are stored, thus allow for destruction of cryptographic keys via direct overwrites of key and key material data. Note that keys material stored using storage technologies that do not support direct overwrites of locations and onetime programmable memories are excluded from the requirement to satisfy this SFR.*

¶ 228 **Assurance activity:**

¶ 229 ***TSS:***

¶ 230 The evaluator shall verify the TSS provides a high level description of how keys and key material are destroyed.

¶ 231 ***KMD:***

¶ 232 The evaluator shall check to ensure the KMD lists each type of key material, its origin, possible temporary locations (e.g. key register, cache memory, stack, FIFO), and storage location.

¶ 233 The evaluator shall verify that the KMD describes when each type of key material is destroyed (for example, on system power off, on wipe function, on disconnection of trusted channels, when no longer needed by the trusted channel per the protocol, etc.).

¶ 234 The evaluator shall also verify that, for each type of key and storage, the type of destruction procedure that is performed (cryptographic erase, overwrite with zeros, overwrite with random pattern, or block erase) is listed. 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 clearing procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are destroyed by overwriting once with zeros, while secret keys stored on the internal persistent storage device are destroyed by overwriting three times with a random pattern that is changed before each write").

¶ 235 The evaluator shall check to ensure the KMD lists each type of key material (software-based key storage, BEVs, passwords, etc.) and its origin, storage location, and the method for destruction for each key.

¶ 236 ***Test:***

¶ 237 For each software and firmware key destruction situation the evaluator shall repeat the following tests for Nonvolatile Memory. There is no test for keys in volatile memory, since they are destroyed by powering down the TOE. For the test below, “key” refers to keys and key material.

¶ 238 Test 1: The evaluator shall utilize appropriate combinations of specialized Operational Environment (e.g. a Virtual Machine) and development tools

(debuggers, simulators, etc.) to test that keys are destroyed, including all copies of the key that may have been created internally by the TOE during normal cryptographic processing with that key.

¶ 239 For each key subject to destruction, including intermediate copies of keys that are persisted encrypted by the TOE the evaluator shall:

1. Attach to the TOE software/firmware with a debugger, or use alternative methods to perform the tests that follow, including the use of developer-provided special tools that allow inspection of device memory in a special test configuration.
2. Record the value of the key in the TOE subject to destruction.
3. Cause the TOE to perform a normal cryptographic processing with the key from #1.
4. Cause the TOE to destroy the key.
5. Cause the TOE to stop the execution but not exit.
6. Cause the TOE to dump the entire memory footprint of the TOE into a binary file.
7. Search the content of the binary file created in #6 for instances of the known key value from #2.

¶ 240 The test succeeds if no copies of the key from #2 are found in step #7 above and fails otherwise.

¶ 241 The evaluator shall perform this test on all keys subject to destruction, including those persisted in encrypted form, to ensure intermediate copies are cleared.

4.5.5 FCS_COP.1(a) Cryptographic Operation (Symmetric encryption/decryption)

(for O.COMMS_PROTECTION)

Hierarchical to: No other components.

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(b) Cryptographic key generation (Symmetric Keys)]

FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

¶ 242 **FCS_COP.1.1(a) Refinement**: The TSF shall perform **encryption and decryption** in accordance with a specified cryptographic algorithm **AES operating in [assignment: *one or more modes*]** and cryptographic key sizes **128-bits and 256-bits** that meets the following:

- **FIPS PUB 197, "Advanced Encryption Standard (AES)"**
- **[Selection: *NIST SP 800-38A, NIST SP 800-38B, NIST SP 800-38C, NIST SP 800-38D*]**

¶ 243 ***Application Note:***

¶ 244 *For the assignment, the ST author should assign the mode or modes in which AES operates to support the cryptographic protocols chosen for FTP_ITC and FTP_TRP.*

¶ 245 *For the selection, the ST author should choose the standards that describe the modes specified in the assignment.*

¶ 246 **Assurance Activity:**

¶ 247 ***Test:***

¶ 248 The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", The CMAC Validation System (CMACVS)", "The Counter with Cipher Block Chaining-Message Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

4.5.6 FCS_COP.1(b) Cryptographic Operation (for signature generation/verification)

(for O.UPDATE_VERIFICATION, O.COMMS_PROTECTION)

Hierarchical to: No other components.

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_EXT.4 Extended: Cryptographic Key Material Destruction

¶ 249 **FCS_COP.1.1(b) Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a **[selection**:

- ***Digital Signature Algorithm (DSA) with key sizes (modulus) of* [assignment: *2048 bits or greater*],**
- ***RSA Digital Signature Algorithm (rDSA) with key sizes (modulus) of* [assignment: *2048 bits or greater*], *or***
- ***Elliptic Curve Digital Signature Algorithm (ECDSA) with key sizes of* [assignment: *256 bits or greater*]]**

¶ 250 that meets the following **[selection**:

¶ 251 Case: Digital Signature Algorithm

- FIPS PUB 186-4, “Digital Signature Standard”

¶ 252 Case: RSA Digital Signature Algorithm

- FIPS PUB 186-4, “Digital Signature Standard”

¶ 253 Case: Elliptic Curve Digital Signature Algorithm

- FIPS PUB 186-4, “Digital Signature Standard”
- The TSF shall implement “NIST curves” P-256, P384 and [selection: P521, no other curves] (as defined in FIPS PUB 186-4, “Digital Signature Standard”).

¶ 254].

¶ 255 ***Application Note:***

¶ 256 *The ST Author should choose the algorithm implemented to perform digital signatures; if more than one algorithm is available, this requirement (and the corresponding FCS_CKM.1 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.

¶ 257 *For elliptic curve-based schemes, the key size refers to the log_2 of the order of the base point.*

¶ 258 **Assurance Activity:**

¶ 259 ***Test:***

¶ 260 The evaluator shall use the signature generation and signature verification portions of "The Digital Signature Algorithm Validation System" (DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System" (ECDSA2VS), and "The RSA Validation System" RSA2VS as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e., FIPS PUB 186-4). This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.

4.5.7 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

(for O.STORAGE_ENCRYPTION and O.COMMS_PROTECTION)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 261 **FCS_RBG_EXT.1.1:** The TSF shall perform all deterministic random bit generation services in accordance with [selection: *ISO/IEC 18031:2011, NIST SP 800-90A*] using [selection: *Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)*].

¶ 262 **FCS_RBG_EXT.1.2** The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: [assignment: *number of software-based sources*] software-based noise source(s), [assignment: *number of hardware-based sources*] hardware-based noise source(s)] with a minimum of [selection: *128 bits, 256 bits*] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

¶ 263 ***Application Note:***

¶ 264 *ISO/IEC 18031:2011 contains 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 and include the specific underlying cryptographic primitives used in the requirement. 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 CTR_DRBG are allowed. Table C.2 in ISO/IEC 18031:2011 provides an identification of Security strengths, Entropy and Seed length requirements for the AES-128 and 256 Block Cipher.*

¶ 265 *The CTR_DRGB in ISO/IEC 18031:2011 requires using derivation function, whereas NIST SP 800-90A does not. Either model is acceptable. In the first selection in FCS_RBG_EXT.1.1, the ST Author chooses the standard with which they are compliant.*

¶ 266 *The first selection in FCS_RBG_EXT.1.2 the ST author fills in how many entropy sources are used for each type of entropy source they employ. It should be noted that a combination of hardware and software based noise sources is acceptable.*

¶ 267 *It should be noted that the entropy source is considered to be a part of the RBG and if the RBG is included in the TOE, the developer is required to provide the entropy description outlined in Appendix E. The documentation *and tests* required in the Evaluation Activity for this element necessarily cover each source indicated in FCS_RBG_EXT.1.2.*

¶ 268 **Assurance activity:**

¶ 269 ***TSS:***

¶ 270 For any RBG services provided by a third party, the evaluator shall ensure the TSS includes a statement about the expected amount of entropy received from such a source, and a full description of the processing of the output of the third-party source. The evaluator shall verify that this statement is consistent with the selection made in FCS_RBG_EXT.1.2 for the seeding of the DRBG. If the ST specifies more than one DRBG, the evaluator shall examine the TSS to verify that it identifies the usage of each DRBG mechanism.

¶ 271 ***Entropy Description:***

¶ 272 The evaluator shall ensure the Entropy Description provides all of the required information as described in Appendix E. The evaluator assesses the information provided and ensures the TOE is providing sufficient entropy when it is generating a Random Bit String.

¶ 273 ***Operational Guidance:***

¶ 274 The evaluator shall verify that the AGD guidance instructs the administrator how to configure the TOE to use the selected DRBG mechanism(s), if necessary.

¶ 275 ***Test:***

¶ 276 The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable by the TOE, the evaluator shall perform 15 trials for each configuration. The evaluator shall verify that the instructions in the operational guidance for configuration of the RBG are valid.

¶ 277 If the RBG 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 SP800-90A).

¶ 278 If the RBG 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.

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

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

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

¶ 282 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.

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

4.6 Class FDP: User Data Protection

¶ 284 ***Application Note:***

¶ 285 *The User Data Access Control SFP is composed of Table 2, Table 3, FDP_ACC.1, FDP_ACF.1, FMT_MSA.1, and FMT_MSA.3.*

4.6.1 FDP_ACC.1 Subset access control

(for O.ACCESS_CONTROL and O.USER_AUTHORIZATION)

Hierarchical to: No other components.

Dependencies: FDP_ACF.1 Security attribute based access control

¶ 286 **FDP_ACC.1.1 Refinement:** The TSF shall enforce the **User Data Access Control SFP** on subjects, objects, and operations among subjects and objects specified in **Table 2 and Table 3.**

¶ 287 **Assurance Activity:**

¶ 288 It is covered by assurance activities for FDP_ACF.1.

4.6.2 FDP_ACF.1 Security attribute based access control

(for O.ACCESS_CONTROL and O.USER_AUTHORIZATION)

Hierarchical to: No other components.

Dependencies: FDP_ACC.1 Subset access control

FMT_MSA.3 Static attribute initialization

¶ 289 **FDP_ACF.1.1 Refinement:** The TSF shall enforce the **User Data Access Control SFP** to objects based on the following: subjects, objects, and attributes specified in **Table 2 and Table 3.**

¶ 290 **FDP_ACF.1.2 Refinement:** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: ***rules governing access among controlled subjects and controlled objects using controlled operations on controlled objects specified in Table 2 and Table 3***.

¶ 291 **FDP_ACF.1.3 Refinement:** The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [assignment: *rules* ***that do not conflict with the User Data Access Control SFP****, based on security attributes, that explicitly authorise access of subjects to objects*].

¶ 292 **FDP_ACF.1.4 Refinement:** The TSF shall explicitly deny access of subjects to objects based on the following additional rules: [assignment: *rules* ***that do not conflict with the User Data Access Control SFP****, based on security attributes, that explicitly deny access of subjects to objects*].

Table 2 D.USER.DOC Access Control SFP

		"Create"	"Read"	"Modify"	"Delete"
Print	***Operation:***	***Submit a document to be printed***	***View image or Release printed output***	***Modify stored document***	***Delete stored document***
	Job owner	(note 1)			
	U.ADMIN				
	U.NORMAL		denied	denied	denied
	Unauthenticated	(condition 1)	denied	denied	denied

		"Create"	"Read"	"Modify"	"Delete"
Scan	***Operation:***	***Submit a document for scanning***	***View scanned image***	***Modify stored image***	***Delete stored image***
	Job owner	(note 2)			
	U.ADMIN				
	U.NORMAL		denied	denied	denied
	Unauthenticated	denied	denied	denied	denied
Copy	***Operation:***	***Submit a document for copying***	***View scanned image or Release printed copy output***	***Modify stored image***	***Delete stored image***
	Job owner	(note 2)			
	U.ADMIN				
	U.NORMAL		denied	denied	denied
	Unauthenticated	denied	denied	denied	denied
Fax send	***Operation:***	***Submit a document to send as a fax***	***View scanned image***	***Modify stored image***	***Delete stored image***
	Job owner	(note 2)			
	U.ADMIN				
	U.NORMAL		denied	denied	denied
	Unauthenticated	denied	denied	denied	denied
Fax receive	***Operation:***	***Receive a fax and store it***	***View fax image or Release printed fax output***	***Modify image of received fax***	***Delete image of received fax***
	Fax owner	(note 3)			
	U.ADMIN	(note 4)			
	U.NORMAL	(note 4)	denied	denied	denied
	Unauthenticated		denied	denied	denied
Storage / retrieval	***Operation:***	***Store document***	***Retrieve stored document***	***Modify stored document***	***Delete stored document***
	Job owner	(note 1)			
	U.ADMIN				

		"Create"	"Read"	"Modify"	"Delete"
	U.NORMAL		denied	denied	denied
	Unauthenticated	(condition 1)	denied	denied	denied

Table 3 D.USER.JOB Access Control SFP

		"Create" *	"Read"	"Modify"	"Delete"
Print	***Operation:***	***Create print job***	***View print queue / log***	***Modify print job***	***Cancel print job***
	Job owner	(note 1)			
	U.ADMIN				
	U.NORMAL			denied	denied
	Unauthenticated			denied	denied
Scan	***Operation:***	***Create scan job***	***View scan status / log***	***Modify scan job***	***Cancel scan job***
	Job owner	(note 2)			
	U.ADMIN				
	U.NORMAL			denied	denied
	Unauthenticated	denied		denied	denied
Copy	***Operation:***	***Create copy job***	***View copy status / log***	***Modify copy job***	***Cancel copy job***
	Job owner	(note 2)			
	U.ADMIN				
	U.NORMAL			denied	denied
	Unauthenticated	denied		denied	denied
Fax send	***Operation:***	***Create fax send job***	***View fax job queue / log***	***Modify fax send job***	***Cancel fax send job***
	Job owner	(note 2)			
	U.ADMIN				
	U.NORMAL			denied	denied
	Unauthenticated	denied		denied	denied
Fax receive	***Operation:***	***Create fax receive job***	***View fax receive status / log***	***Modify fax receive job***	***Cancel fax receive job***
	Fax owner	(note 3)			
	U.ADMIN	(note 4)			
	U.NORMAL	(note 4)		denied	denied
	Unauthenticated			denied	denied
Storage / retrieval	***Operation:***	***Create storage /***	***View storage /***	***Modify storage /***	***Cancel storage /***

		"Create" *	"Read"	"Modify"	"Delete"
		retrieval job	***retrieval log***	***retrieval job***	***retrieval job***
	Job owner	(note 1)			
	U.ADMIN				
	U.NORMAL			denied	denied
	Unauthenticated	(condition 1)		denied	denied

¶ 293 ***Application note:***

¶ 294 *In general, the ST Author may modify this SFP provided that any changes are more restrictive. As examples, the ST Author may: remove the rules related to Document Processing functions that are not present in a TOE, add or modify rules to further deny access, or subdivide User Data to further restrict access for some data (e.g., D.USER.JOB.PROT and D.USER.JOB.CONF). Empty cells in the table indicate that the operation may be permitted, but it is not required to be permitted.*

¶ 295 *In particular, referring to Table 2 and Table 3:*

- *A cell marked "Denied" indicates that the user (row) must not be permitted to perform the operation (column). The ST Author cannot override this.*
- *A cell that is blank indicates that the user may be permitted to perform the operation. However, the ST author may add conditions or restrictions, or deny permission entirely.*
- *A cell that is marked with a Condition means that the user can be permitted to perform the operation, provided that it meets that Condition as specified below. As with blank cells, the ST author can make it more restrictive.*

¶ 296 ***Condition 1:*** *Jobs submitted by unauthenticated users must contain a credential that the TOE can use to identify the Job Owner.*

¶ 297 *See also the following Notes that are referenced in Table 2 and Table 3:*

¶ 298 ***Note 1****: Job Owner is identified by a credential or assigned to an authorized User as part of the process of submitting a print or storage Job.*

¶ 299 ***Note 2****: Job Owner is assigned to an authorized User as part of the process of initiating a scan, copy, fax send, or retrieval Job.*

¶ 300 ***Note 3****: Job Owner of received faxes is assigned by default or configuration. Minimally, ownership of received faxes is assigned to a specific user or U.ADMIN role.*

¶ 301 ***Note 4****: PSTN faxes are received from outside of the TOE, they are not initiated by Users of the TOE.*

¶ 302 **Assurance Activity:**

¶ 303 ***TSS:***

¶ 304 The evaluator shall check to ensure that the TSS describes the functions to realize SFP defined in Table 2 and Table 3.

¶ 305 ***Operational Guidance:***

¶ 306 The evaluator shall check to ensure that the operational guidance contains a description of the operation to realize the SFP defined in Table 2 and Table 3, which is consistent with the description in the TSS.

¶ 307 ***Test:***

¶ 308 The evaluator shall perform tests to confirm the functions to realize the SFP defined in Table 2 and Table 3 with each type of interface (e.g., operation panel, Web interfaces) to the TOE.

¶ 309 The evaluator testing should include the following viewpoints:

- representative sets of the operations against representative sets of the object types defined in Table 2 and Table 3 (including some cases where operations are either permitted or denied)
- representative sets for the combinations of the setting for security attributes that are used in access control

4.7 Class FIA: Identification and Authentication

4.7.1 FIA_AFL.1 Authentication failure handling

(for O.USER_I&A)

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

¶ 310 **FIA_AFL.1.1** The TSF shall detect when [selection: [assignment: *positive integer number*], an administrator configurable positive integer within [assignment: *range of acceptable values*]] unsuccessful authentication attempts occur related to [assignment: *list of authentication events*].

¶ 311 **FIA_AFL.1.2** When the defined number of unsuccessful authentication attempts has been [selection: met, surpassed], the TSF shall [assignment: *list of actions*].

¶ 312 ***Application note:***

¶ 313 *This SFR applies only to internal identification and authentication.*

¶ 314 **Assurance Activity:**

¶ 315 ***TSS:***

¶ 316 The evaluator shall check to ensure that the TSS contains a description of the actions in the case of authentication failure (types of authentication events, the number of unsuccessful authentication attempts, actions to be conducted), which is consistent with the definition of the SFR.

¶ 317 ***Operational Guidance:***

¶ 318 The evaluator shall check to ensure that the administrator guidance describes the setting for actions to be taken in the case of authentication failure, if any are defined in the SFR.

¶ 319 ***Test:***

¶ 320 The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that the subsequent authentication attempts do not succeed by the behavior according to the actions defined in the SFR when unsuccessful authentication attempts reach the status defined in the SFR.
2. The evaluator shall check to ensure that authentication attempts succeed when conditions to re-enable authentication attempts are defined in the SFR and when the conditions are fulfilled.
3. The evaluator shall perform the tests 1 and 2 described above for all the targeted authentication methods when there are multiple Internal Authentication methods (e.g., password authentication, biometric

authentication).

4. The evaluator shall perform the tests 1 and 2 described above for all interfaces when there are multiple interfaces (e.g., operation panel, Web interfaces) that implement authentication attempts.

4.7.2 FIA_ATD.1 User attribute definition

(for O.USER_AUTHORIZATION)

Hierarchical to: No other components.

Dependencies: No dependencies.

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

¶ 322 ***Application note:***

¶ 323 *The list of security attributes should be the union of all attributes for each of the supported authentication methods.*

¶ 324 **Assurance Activity:**

¶ 325 ***TSS:***

¶ 326 The evaluator shall check to ensure that the TSS contains a description of the user security attributes that the TOE uses to implement the SFR, which is consistent with the definition of the SFR.

4.7.3 FIA_PMG_EXT.1 Extended: Password Management

(for O.USER_I&A)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 327 **FIA_PMG_EXT.1.1** The TSF shall provide the following password management capabilities for User passwords:

- Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [selection: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [assignment: *other characters*]];
- Minimum password length shall be settable by an Administrator, and have the

capability to require passwords of 15 characters or greater;

¶ 328 ***Application Note:***

¶ 329 *This SFR applies only to password-based single-factor Internal Authentication.*

¶ 330 **Assurance Activity:**

¶ 331 ***Operational Guidance:***

¶ 332 The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of passwords, and that it provides instructions on setting the minimum password length.

¶ 333 ***Test:***

¶ 334 The evaluator shall also perform the following test:

¶ 335 The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.

4.7.4 FIA_UAU.1 Timing of authentication

(for O.USER_I&A)

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

¶ 336 **FIA_UAU.1.1 Refinement:** The TSF shall allow [assignment: *list of TSF mediated actions* ***that do not conflict with the User Data Access Control SFP, and do not provide access to D.TSF.CONF, and do not change any TSF data***] on behalf of the user to be performed before the user is authenticated.

¶ 337 **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.

¶ 338 ***Application note:***

¶ 339 *User authentication may be performed internally by the TOE or externally by an External IT Entity.*

¶ 340 **Assurance Activity:**

¶ 341 ***TSS:***

¶ 342 The evaluator shall check to ensure that the TSS describes all the identification and authentication mechanisms that the TOE provides (e.g., Internal Authentication and authentication by external servers).

¶ 343 The evaluator shall check to ensure that the TSS identifies all the interfaces to perform identification and authentication (e.g., identification and authentication from operation panel or via Web interfaces).

¶ 344 The evaluator shall check to ensure that the TSS describes the protocols (e.g., LDAP, Kerberos, OCSP) used in performing identification and authentication when the TOE exchanges identification and authentication with External Authentication servers.

¶ 345 The evaluator shall check to ensure that the TSS contains a description of the permitted actions before performing identification and authentication, which is consistent with the definition of the SFR.

¶ 346 ***Operational Guidance:***

¶ 347 The evaluator shall check to ensure that the administrator guidance contains descriptions of identification and authentication methods that the TOE provides (e.g., External Authentication, Internal Authentication) as well as interfaces (e.g., identification and authentication from operation panel or via Web interfaces), which are consistent with the ST (TSS).

¶ 348 ***Test:***

¶ 349 The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that identification and authentication succeeds, enabling the access to the TOE when using authorized data.
2. The evaluator shall check to ensure that identification and authentication fails, disabling the access to the TOE afterwards when using unauthorized data.

¶ 350 The evaluator shall perform the tests described above for each of the authentication methods that the TOE provides (e.g., External Authentication, Internal Authentication) as well as interfaces (e.g., identification and authentication from operation panel or via Web interfaces).

4.7.5 FIA_UAU.7 Protected authentication feedback

(for O.USER_I&A)

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

¶ 351 **FIA_UAU.7.1** The TSF shall provide only [assignment: *list of feedback*] to the user while the authentication is in progress.

¶ 352 ***Application note:***

¶ 353 *FIA_UAU.7 applies only to authentication processes in which the User interacts with the TOE.*

¶ 354 **Assurance Activity:**

¶ 355 ***TSS:***

¶ 356 The evaluator shall check to ensure that the TSS contains a description of the authentication information feedback provided to users while the authentication is in progress, which is consistent with the definition of the SFR.

¶ 357 ***Test:***

¶ 358 The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that only the information defined in the SFR is provided for feedback by attempting identification and authentication.
2. The evaluator shall perform the test 1 described above for all the interfaces that the TOE provides (e.g., operation panel, identification and authentication via Web interface).

4.7.6 FIA_UID.1 Timing of identification

(for O.USER_I&A and O.ADMIN_ROLES)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 359 **FIA_UID.1.1 Refinement:** The TSF shall allow [assignment: *list of TSF-mediated actions **that do not conflict with the User Data Access Control SFP, and do not provide access to D.TSF.CONF, and do not change any TSF data***] on behalf of the user to be performed before the user is identified.

¶ 360 **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.

¶ 361 ***Application note:***

¶ 362 *User identification may be performed internally by the TOE or externally by an External IT Entity.*

¶ 363 **Assurance Activity:**

¶ 364 It is covered by assurance activities for FIA_UAU.1.

4.7.7 FIA_USB.1 User-subject binding

(for O.USER_I&A)

Hierarchical to: No other components.

Dependencies: FIA_ATD.1 User attribute definition

¶ 365 **FIA_USB.1.1** The TSF shall associate the following user security attributes with subjects acting on the behalf of that user: [assignment: *list of user security attributes*].

¶ 366 **FIA_USB.1.2** The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users: [assignment: *rules for the initial association of attributes*].

¶ 367 **FIA_USB.1.3** The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users: [assignment: *rules for the changing of attributes*].

¶ 368 **Assurance Activity:**

¶ 369 ***TSS:***

¶ 370 The evaluator shall check to ensure that the TSS contains a description of rules

for associating security attributes with the users who succeed identification and authentication, which is consistent with the definition of the SFR.

¶ 371 ***Test:***

¶ 372 The evaluator shall also perform the following test:

¶ 373 The evaluator shall check to ensure that security attributes defined in the SFR are associated with the users who succeed identification and authentication (it is ensured in the tests of FDP_ACF) for each role that the TOE supports (e.g., User and Administrator).

4.8 Class FMT: Security Management

4.8.1 FMT_MOF.1 Management of security functions behavior

(for O.ADMIN_ROLES)

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

¶ 374 **FMT_MOF.1.1 Refinement:** The TSF shall restrict the ability to [selection: *determine the behaviour of, disable, enable, modify the behaviour of*] the functions [assignment: *list of functions*] to **U.ADMIN**.

¶ 375 **Assurance Activity:**

¶ 376 ***TSS:***

¶ 377 The evaluator shall check to ensure that the TSS contains a description of the management functions that the TOE provides as well as user roles that are permitted to manage the functions, which is consistent with the definition of the SFR.

¶ 378 The evaluator shall check to ensure that the TSS identifies interfaces to operate the management functions.

¶ 379 ***Operational Guidance:***

¶ 380 The evaluator shall check to ensure that the administrator guidance describes the operation methods for users of the given roles defined in the SFR to operate the

management functions.

¶ 381 ***Test:***

¶ 382 The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that users of the given roles defined in the SFR can operate the management functions in accordance with the operation methods specified in the administrator guidance.
2. The evaluator shall check to ensure that the operation results are appropriately reflected.
3. The evaluator shall check to ensure that U.NORMAL is not permitted to operate the management functions.

4.8.2 FMT_MSA.1 Management of security attributes

(for O.ACCESS_CONTROL and O.USER_AUTHORIZATION)

Hierarchical to: No other components.

Dependencies: [FDP_ACC.1 Subset access control, ~~or~~ ~~FDP_IFC.1 Subset information flow control~~]

FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

¶ 383 **FMT_MSA.1.1 Refinement:** The TSF shall enforce the **User Data Access Control SFP** to restrict the ability to [selection: *change_default, query, modify, delete, [assignment: other operations]*] the security attributes [assignment: *list of security attributes*] to [assignment: *the authorised identified roles*].

¶ 384 **Assurance Activity:**

¶ 385 ***TSS:***

¶ 386 The evaluator shall check to ensure that the TSS contains a description of possible operations for security attributes and given roles to those security attributes, which is consistent with the definition of the SFR.

¶ 387 ***Operational Guidance:***

¶ 388 The evaluator shall check to ensure that the administrator guidance contains a

description of possible operations for security attributes and given roles to those security attributes, which is consistent with the definition of the SFR.

¶ 389 The evaluator shall check to ensure that the administrator guidance describes the timing of modified security attributes.

¶ 390 ***Test:***

¶ 391 The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that users of the given roles defined in the SFR can perform operations to the security attributes in accordance with the operation methods specified in the administrator guidance.
2. The evaluator shall check to ensure that the operation results are appropriately reflected as specified in the administrator guidance.
3. The evaluator shall check to ensure that a user that is not part of an authorized role defined in the SFR is not permitted to perform operations on the security attributes.

4.8.3 FMT_MSA.3 Static attribute initialization

(for O.ACCESS_CONTROL and O.USER_AUTHORIZATION)

Hierarchical to: No other components.

Dependencies: FMT_MSA.1 Management of security attributes

FMT_SMR.1 Security roles

¶ 392 **FMT_MSA.3.1 Refinement:** The TSF shall enforce the **User Data Access Control SFP** to provide [selection, choose one of: *restrictive, permissive, [assignment: other property]*] default values for security attributes that are used to enforce the SFP.

¶ 393 **FMT_MSA.3.2 Refinement:** The TSF shall allow the **[selection: *U.ADMIN, no role*]** to specify alternative initial values to override the default values when an object or information is created.

¶ 394 ***Application note:***

¶ 395 *FMT_MSA.3.2 applies only to security attributes whose default values can be overridden.*

¶ 396 **Assurance Activity:**

¶ 397 ***TSS:***

¶ 398 The evaluator shall check to ensure that the TSS describes mechanisms to generate security attributes which have properties of default values, which are defined in the SFR.

¶ 399 ***Test:***

¶ 400 If U.ADMIN is selected, then testing of this SFR is performed in the tests of FDP_ACF.1.

4.8.4 FMT_MTD.1 Management of TSF data

(for O.ACCESS CONTROL)

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

¶ 401 **FMT_MTD.1.1 Refinement:** The TSF shall restrict the ability to **perform the specified operations on the specified TSF Data to the roles specified in Table 4.**

Table 4 Management of TSF Data

Data	Operation	Authorised role(s)
[assignment: *list of TSF Data owned by a U.NORMAL or associated with Documents or jobs owned by a U.NORMAL*]	[selection: *change default, query, modify, delete, clear,* [assignment: *other operations*]]	U.ADMIN, the owning U.NORMAL.
[assignment: *list of TSF Data not owned by a U.NORMAL*]	[selection: *change default, query, modify, delete, clear,* [assignment: *other operations*]]	U.ADMIN
[assignment: *list of software, firmware, and related configuration data*]	[selection: *change default, query, modify, delete, clear,* [assignment: *other operations*]]	U.ADMIN

¶ 402 **Assurance Activity:**

¶ 403 ***Operational Guidance:***

¶ 404 The evaluator shall check to ensure that the administrator guidance identifies the management operations and authorized roles consistent with the SFR.

¶ 405 The evaluator shall check to ensure that the administrator guidance describes how the assignment of roles is managed.

¶ 406 The evaluator shall check to ensure that the administrator guidance describes how security attributes are assigned and managed.

¶ 407 The evaluator shall check to ensure that the administrator guidance describes how the security-related rules (.e.g., access control rules, timeout, number of consecutive logon failures,) are configured.

¶ 408 ***Test:***

¶ 409 The evaluator shall perform the following tests:

1. The evaluator shall check to ensure that users of the given roles defined in the SFR can perform operations to TSF data in accordance with the operation methods specified in the administrator guidance.
2. The evaluator shall check to ensure that the operation results are appropriately reflected as specified in the administrator guidance.
3. The evaluator shall check to ensure that no users other than users of the given roles defined in the SFR can perform operations to TSF data.

4.8.5 FMT_SMF.1 Specification of Management Functions

(for O.USER_AUTHORIZATION, O.ACCESS_CONTROL, and O.ADMIN_ROLES)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 410 **FMT_SMF.1.1 Refinement:** The TSF shall be capable of performing the following management functions: [assignment: *list of management functions provided by the TSF*].

¶ 411 ***Application note:***

¶ 412 *Regarding "management functions provided by the TSF", the ST Author should consider management functions that support the security objectives of this protection profile.*

¶ 413 *The management functions should be restricted to the authorized identified role in FMT_MOF.1, FMT_MTD.1, FMT_MSA.1.*

¶ 414 *The ST Author may identify cases where a security objective is fulfilled without explicit manageability.*

¶ 415 *For example, the following management functions are categorized by security objectives:*

¶ 416 *For O.USER_AUTHORIZATION, O.USER_I&A, O.ADMIN_ROLES, O.ACCESS_CONTROL:*

- *User management (e.g., add/change/remove local user)*
- *Role management (e.g., assign/deassign role relationship with user)*
- *Configuring identification and authentication (e.g., selecting between local and external I&A)*
- *Configuring authorization and access controls (e.g., access control lists for TOE resources)*
- *Configuring communication with External IT Entities*

¶ 417 *For O.UPDATE_VERIFICATION:*

- *Configuring software updates*

¶ 418 *For O.COMMS_PROTECTION:*

- *Configuring network communications*
- *Configuring the system or network time source*

¶ 419 *For O.AUDIT:*

- *Configuring data transmission to audit server*
- *Configuring the system or network time source*
- *Configuring internal audit log storage*

¶ 420 *For O.STORAGE_ENCRYPTION, O.KEY_MATERIAL:*

- *Configuring and invoking encryption of Field-Replaceable Nonvolatile Storage Devices*

¶ 421 *(Optional) For O.IMAGE_OVERWRITE, O.PURGE DATA:*

- *Configuring and/or invoking image overwrite functions*
- *Configuring and/or invoking data purging functions*

¶ 422 **Assurance Activity:**

¶ 423 ***TSS:***

¶ 424 The evaluator shall check the TSS to ensure that the management functions are consistent with the assignment in the SFR.

¶ 425 ***Operational Guidance:***

¶ 426 The evaluator shall check the guidance documents to ensure that management functions are consistent with the assignment in the SFR, and that their operation is described.

4.8.6 FMT_SMR.1 Security roles

(for O.ACCESS_CONTROL, O.USER_AUTHORIZATION, and O.ADMIN_ROLES)

Hierarchical to: No other components.

Dependencies: FIA_UID.1 Timing of identification

¶ 427 **FMT_SMR.1.1 Refinement:** The TSF shall maintain the roles **U.ADMIN, U.NORMAL**.

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

¶ 429 **Assurance Activity:**

¶ 430 ***TSS:***

¶ 431 The evaluator shall check to ensure that the TSS contains a description of security related roles that the TOE maintains, which is consistent with the definition of the SFR.

¶ 432 ***Test:***

¶ 433 As for tests of this SFR, it is performed in the tests of FMT_MOF.1, FMT_MSA.1, and FMT_MTD.1.

4.9 Class FPR: Privacy

¶ 434 There are no class FPR requirements.

4.10 Class FPT: Protection of the TSF

4.10.1 FPT_SKP_EXT.1 Extended: Protection of TSF Data

(for O.COMMS_PROTECTION)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 435 **FPT_SKP_EXT.1.1** The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

¶ 436 ***Application Note:***

¶ 437 *The intent of the requirement is that an administrator is unable to read or view the identified keys (stored or ephemeral) through "normal" interfaces. While it is understood that the administrator could directly read memory to view these keys, doing so is not a trivial task and may require substantial work on the part of an administrator. Since the administrator is considered a trusted agent, it is assumed they would not engage in such an activity.*

¶ 438 **Assurance Activity:**

¶ 439 ***TSS:***

¶ 440 The evaluator shall examine the TSS to determine that it details how any pre-shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.

4.10.2 FPT_STM.1 Reliable time stamps

(for.O.AUDIT)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 441 **FPT_STM.1.1** The TSF shall be able to provide reliable time stamps.

¶ 442 ***Application note:***

¶ 443 *The time may be set by a trusted administrator or by a network service (e.g., NTP) from a trusted External IT Entity.*

¶ 444 **Assurance Activity:**

¶ 445 ***TSS:***

¶ 446 The evaluator shall check to ensure that the TSS describes mechanisms that provide reliable time stamps.

¶ 447 ***Operational Guidance:***

¶ 448 The evaluator shall check to ensure that the guidance describes the method of setting the time.

¶ 449 ***Test:***

¶ 450 The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that the time is correctly set up in accordance with the guidance or external network services (e.g., NTP).
2. The evaluator shall check to ensure that the time stamps are appropriately provided.

4.10.3 FPT_TST_EXT.1 Extended: TSF testing

(for O.TSF_SELF_TEST)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 451 **FPT_TST_EXT.1.1** The TSF shall run a suite of self-tests during initial start-up (and power on) to demonstrate the correct operation of the TSF.

¶ 452 ***Application note:***

¶ 453 *Power-on self-tests may take place before the TSF is operational, in which case this SFR can be satisfied by verifying the TSF image by digital signature as specified in FCS_COP.1(b), or by hash specified in FCS_COP.1(c).*

¶ 454 **Assurance Activity:**

¶ 455 ***TSS:***

¶ 456 The evaluator shall examine the TSS to ensure that it details the self-tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly.

¶ 457 ***Operational Guidance:***

¶ 458 The evaluator shall also ensure that the operational guidance describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.

4.10.4 FPT_TUD_EXT.1 Extended: Trusted Update

(for O.UPDATE_VERIFICATION)

Hierarchical to: No other components.

Dependencies: [FCS_COP.1(b) Cryptographic Operation (for signature generation/verification), or

FCS_COP.1(c) Cryptographic operation (Hash Algorithm)].

¶ 459 **FPT_TUD_EXT.1.1** The TSF shall provide authorized administrators the ability to query the current version of the TOE firmware/software.

¶ 460 **FPT_TUD_EXT.1.2** The TSF shall provide authorized administrators the ability to initiate updates to TOE firmware/software.

¶ 461 **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.

¶ 462 ***Application note:***

¶ 463 *FPT_TUD_EXT.1.2 may be interpreted to allow an administrator to "pre-authorize" automatic updates, provided that they are verified according to FPT_TUD_EXT.1.3.*

¶ 464 *The digital signature mechanism is specified in FCS_COP.1(b). The published hash is generated by one of the functions specified in FCS_COP.1(c). It is acceptable to implement both mechanisms.*

¶ 465 **Assurance Activity:**

¶ 466 ***TSS:***

¶ 467 The evaluator shall check to ensure that the TSS contains a description of mechanisms that verify software for update when performing updates, which is consistent with the definition of the SFR.

¶ 468 The evaluator shall check to ensure that the TSS identifies interfaces for administrators to obtain the current version of the TOE as well as interfaces to perform updates.

¶ 469 ***Operational Guidance:***

¶ 470 The evaluator shall check to ensure that the administrator guidance contains descriptions of the operation methods to obtain the TOE version as well as the operation methods to start update processing, which are consistent with the description of the TSS.

¶ 471 ***Test:***

¶ 472 The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure the current version of the TOE can be appropriately obtained by means of the operation methods specified by the administrator guidance.
2. The evaluator shall check to ensure that the verification of the data for updates of the TOE succeeds using authorized data for updates by means of the operation methods specified by the administrator guidance.
3. The evaluator shall check to ensure that only administrators can implement the application for updates using authorized data for updates.

4. The evaluator shall check to ensure that the updates are correctly performed by obtaining the current version of the TOE after the normal updates finish.

5. The evaluator shall check to ensure that the verification of the data for updates of the TOE fails using unauthorized data for updates by means of the operation methods specified by the administrator guidance. (The evaluator shall also check those cases where hash verification mechanism and digital signature verification mechanism fail.)

4.11 Class FRU: Resource Utilization

¶ 473 There are no class FRU requirements.

4.12 Class FTA: TOE Access

4.12.1 FTA_SSL.3 TSF-initiated termination

(for O.USER_I&A)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 474 **FTA_SSL.3.1** The TSF shall terminate an interactive session after a [assignment: *time interval of user inactivity*].

¶ 475 **Assurance Activity:**

¶ 476 ***TSS:***

¶ 477 The evaluator shall check to ensure that the TSS describes the types of user sessions to be terminated (e.g., user sessions via operation panel or Web interfaces) after a specified period of user inactivity.

¶ 478 ***Operational Guidance:***

¶ 479 The evaluator shall check to ensure that the guidance describes the default time interval and, if it is settable, the method of setting the time intervals until the termination of the session.

¶ 480 ***Test:***

¶ 481 The evaluator shall also perform the following tests:

1. If it is settable, the evaluator shall check to ensure that the time until the termination of the session can be set up by the method of setting specified in the administrator guidance.
2. The evaluator shall check to ensure that the session terminates after the specified time interval.
3. The evaluator shall perform the tests 1 and 2 described above for all the user sessions identified in the TSS.

4.13 Class FTP: Trusted Paths/Channels

4.13.1 FTP_ITC.1 Inter-TSF trusted channel

(for O.COMMS_PROTECTION, O.AUDIT)

Hierarchical to: No other components.

Dependencies: [FCS_IPSEC_EXT.1 Extended: IPsec selected, or
FCS_TLS_EXT.1 Extended: TLS selected, or
FCS_SSH_EXT.1 Extended: SSH selected, or
FCS_HTTPS_EXT.1 Extended: HTTPS selected].

¶ 482 **FTP_ITC.1.1 Refinement:** The TSF shall **use [selection: *IPsec, SSH, TLS, TLS/HTTPS*] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: [selection: *authentication server, [assignment: other capabilities]*]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data**.

¶ 483 **FTP_ITC.1.2 Refinement:** The TSF shall permit the TSF, **or the authorized IT entities**, to initiate communication via the trusted channel

¶ 484 **FTP_ITC.1.3 Refinement:** The TSF shall initiate communication via the trusted channel for [assignment: ***list of services for which the TSF is able to initiate communications***].

¶ 485 ***Application note:***

¶ 486 *The assignment in FTP_ITC.1.3 should address the confidentiality and/or integrity requirements for communication of User and TSF Data between the TOE and another IT entity. FTP_TRP.1 is intended to be used for interactive communication between the TOE and remote users.*

¶ 487 *The intent of the above requirement is to use a cryptographic protocol to protect external communications with authorized IT entities that the TOE interacts with to perform its functions. Protection (by one of the listed protocols) is required at least for communications with the server that collects the audit information. If it communicates with an authentication server (e.g., RADIUS), then the ST author chooses "authentication server" in FTP_ITC.1.1 and this connection must be protected by one of the listed protocols. If other authorized IT entities (e.g., NTP server) are protected, the ST author makes the appropriate assignments (for those entities) and selections (for the protocols that are used to protect those connections). After the ST author has made the selections, they are to select the detailed requirements in Appendix D.2 corresponding to their protocol selection to put in the ST. To summarize, the connection to an external audit collection server is required to be protected by one of the listed protocols. If an External Authentication server is supported, then it is required to protect that connection with one of the listed protocols. For any other external server, external communications are not required to be protected, but if protection is claimed, then it must be protected with one of the identified protocols.*

¶ 488 *While there are no requirements on the party initiating the communication, the ST author lists in the assignment for FTP_ITC.1.3 the services for which the TOE can initiate the communication with the authorized IT entity.*

¶ 489 *The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage. It may be the case that some part of the TOE setup involves manually setting up tunnels to protect other communication, and if after an outage the TOE attempts to re-establish the communication automatically with (the necessary) manual intervention, there may be a window created where an attacker might be able to gain critical information or compromise a connection.*

¶ 490 **Assurance Activity:**

¶ 491 ***TSS:***

¶ 492 The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken.

¶ 493 ***Test:***

¶ 494 The evaluator shall also perform the following tests:

1. The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
2. For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE.
3. The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext.
4. The evaluator shall ensure, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.

¶ 495 Further assurance activities are associated with the specific protocols.

4.13.2 FTP_TRP.1(a) Trusted path (for Administrators)

(for O.COMMS_PROTECTION)

Hierarchical to: No other components.

Dependencies: [FCS_IPSEC_EXT.1 Extended: IPsec selected, or

FCS_TLS_EXT.1 Extended: TLS selected, or

FCS_SSH_EXT.1 Extended: SSH selected, or

FCS_HTTPS_EXT.1 Extended: HTTPS selected].

¶ 496 **FTP_TRP.1.1(a) Refinement**: The TSF shall **use** [**selection, choose at least one of: IPsec, SSH, TLS, TLS/HTTPS] to** provide **a trusted** communication path between itself and remote **administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **disclosure and detection of modification of the communicated data**.

¶ 497 **FTP_TRP.1.2(a) Refinement**: The TSF shall permit **remote administrators** to initiate communication via the trusted path

¶ 498 **FTP_TRP.1.3(a) Refinement:** The TSF shall require the use of the trusted path for **initial administrator authentication and all remote administration actions**.

¶ 499 ***Application Note:***

¶ 500 *This requirement ensures that authorized remote administrators initiate all communication with the TOE via a trusted path, and that all communications with the TOE by remote administrators is performed over this path. The data passed in this trusted communication path are encrypted as defined the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix D.2 corresponding to their selection are copied to the ST if not already present.*

¶ 501 **Assurance Activity:**

¶ 502 ***TSS:***

¶ 503 The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST.

¶ 504 ***Operational Guidance:***

¶ 505 The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method.

¶ 506 ***Test:***

¶ 507 The evaluator shall also perform the following tests:

1. The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
2. For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path.
3. The evaluator shall ensure, for each method of remote administration, the channel data are not sent in plaintext.

¶ 508 Further assurance activities are associated with the specific protocols.

4.13.3 FTP_TRP.1(b) Trusted path (for Non-administrators)

(for O.COMMS_PROTECTION)

Hierarchical to: No other components.

Dependencies: [FCS_IPSEC_EXT.1 Extended: IPsec selected, or
FCS_TLS_EXT.1 Extended: TLS selected, or
FCS_SSH_EXT.1 Extended: SSH selected, or
FCS_HTTPS_EXT.1 Extended: HTTPS selected].

¶ 509 **FTP_TRP.1.1(b) Refinement** : The TSF shall **use** [**selection, choose at least one of: IPsec, SSH, TLS, TLS/HTTPS] to** provide **a** trusted communication path between itself and remote users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from **disclosure and detection of modification of the communicated data**.

¶ 510 **FTP_TRP.1.2(b) Refinement**: The TSF shall permit [selection: **the TSF, remote users**] to initiate communication via the trusted path

¶ 511 **FTP_TRP.1.3(b) Refinement**: The TSF shall require the use of the trusted path for **initial user authentication and all remote user actions**.

¶ 512 ***Application Note:***

¶ 513 *This requirement ensures that authorized remote users initiate all communication with the TOE via a trusted path, and that all communications with the TOE by remote users is performed over this path. The data passed in this trusted communication path are encrypted as defined the protocol chosen in the first selection. The ST author chooses the mechanism or mechanisms supported by the TOE, and then ensures the detailed requirements in Appendix D.2 corresponding to their selection are copied to the ST if not already present.*

¶ 514 **Assurance Activity:**

¶ 515 ***TSS:***

¶ 516 The evaluator shall examine the TSS to determine that the methods of remote TOE access for non-administrative users are indicated, along with how those communications are protected.

¶ 517 The evaluator shall also confirm that all protocols listed in the TSS in support of remote TOE access are consistent with those specified in the requirement, and are included in the requirements in the ST.

¶ 518 ***Operational Guidance:***

¶ 519 The evaluator shall confirm that the operational guidance contains instructions for establishing the remote user sessions for each supported method.

¶ 520 ***Test:***

¶ 521 The evaluator shall also perform the following tests:

1. The evaluators shall ensure that communications using each specified (in the operational guidance) remote user access method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful.
2. For each method of remote access supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote user session without invoking the trusted path.
3. The evaluator shall ensure, for each method of remote user access, the channel data are not sent in plaintext.

¶ 522 Further assurance activities are associated with the specific protocols.

4.14 Security Functional Requirements rationale

¶ 523 The dependencies for the SFRs in this PP will differ in some instances from those that are contained in the Common Criteria V3.1, Part 2.

¶ 524 For this document a careful review was performed to assure that the dependencies for the SFRs are consistent with, and appropriate for, the use-cases and threat scenarios that are defined for this class of products. Additionally, the SFR dependencies were reviewed in order to be consistent with the refinements, iterations, and extended requirements defined by the PP. As a result, the dependencies for some of the SFRs are not the same as those identified in the CC.

¶ 525 Note that the only operations performed on the SFRs (refinements, iterations, and pre-completed selections and assignments) are in strict accordance with the allowed operations, as defined in the CC. These operations in some cases cause the SFRs to have additional dependencies that were not defined in CC Part 2. The authors felt that bringing the dependencies in line with the needs of the product class and with the operations that were completed on the SFRs would avoid confusion and the introduction of unnecessary or inconsistent security functions.

¶ 526 Dependencies that have been removed from SFRs are indicated by ~~strikethrough~~ typeface. The dependent SFRs are not present in this PP, so they cannot be used. As examples:

- Dependency on FPT_ITT.1 was removed from communication protocol SFRs because HCDs are not, for the purposes of this PP, distributed TOEs.
- Dependency on either FDP_ITC.1 or FDP_ITC.2 was removed from cryptography SFRs because those mechanisms are not used in the HCD PP.

5 Security Assurance Requirements (APE_REQ)

¶ 527 This section describes Security Assurance Requirements (SARs) in the evaluations performed by the evaluator based on the CC. These are all common to the Security Functional Requirements (SFRs) in Section 4, Appendix B , Appendix C , and Appendix D. Assurance activities to the individual SFRs are described in their respective sections.

¶ 528 After the ST has been approved for evaluation, the Common Criteria IT Security Evaluation Facilities (ITSEF) will obtain the TOE, necessary IT environment, and the TOE guidance documents. The assurance activities described in the ST (which will be refined by the ITSEF to be TOE-specific, either within the ST or in a separate document) will be performed by the ITSEF. Although these activities were performed under the control of the ITSEF, it is allowed to obtain supports from the developer as well. The results of these activities will be documented and presented (along with the administrative guidance used) for validation.

¶ 529 For each assurance 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.

¶ 530 The TOE security assurance requirements specified in Table 5 provides evaluative activities required to address the threats identified in Section 2.3 of this PP.

Table 5 TOE Security Assurance Requirements

Assurance Class	Assurance Components	Assurance Components Description
Security Target Evaluation	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic functional specification
Guidance Documents	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures

Assurance Class	Assurance Components	Assurance Components Description
Life-cycle support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests	ATE_IND.1	Independent testing – Conformance
Vulnerability assessment	AVA_VAN.1	Vulnerability survey

5.1 Class ASE: Security Target evaluation

¶ 531 The ST is evaluated as per ASE activities defined in the CEM. In addition, there may be Assurance Activities specified within the PP that call necessary descriptions to be included in the TSS that are specific to the TOE technology type.

¶ 532 Appendix E provides a description of the information expected to be provided regarding the quality of entropy in the random bit generator.

¶ 533 Given the criticality of the key management scheme, this PP requires the developer to provide a detailed description of their key management implementation. This information can be submitted as an appendix to the ST and marked proprietary, as this level of detailed information is not expected to be made publicly available. See Appendix F for details on the expectation of the developer's Key Management Description.

5.2 Class ADV: Development

¶ 534 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, Appendix B , Appendix C , and Appendix D should provide the ST authors with sufficient information to determine the appropriate content for the TSS section.

5.2.1 ADV_FSP.1 Basic functional specification

¶ 535 The functional specification describes the TSF Interfaces (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. 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:

¶ 536 ADV_FSP.1.1D — The developer shall provide a functional specification.

¶ 537 ADV_FSP.1.2D — The developer shall provide a tracing from the functional specification to the SFRs.

¶ 538 Developer Note: The developer shall provide appropriate TSS description and guidance documents as the functional specification. The TSS description identifies TSFIs associated with each SFR in order to confirm the validity of interface design. The developer is required to provide a description at least at a confirmable level in which TSS description and contents of guidance documents are consistent with each other. In case of insufficient information for evaluation in TSS description and contents of guidance documents, additional documentation can be requested. For the SFRs that cannot be directly operated/confirmed from external interfaces, the developer may be requested to provide additional information.

Content and presentation elements:

¶ 539 ADV_FSP.1.1C — The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

¶ 540 ADV_FSP.1.2C — The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

¶ 541 ADV_FSP.1.3C The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

¶ 542 ADV_FSP.1.4C The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

Evaluator action elements:

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

¶ 544 ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

¶ 545 **Assurance activity:**

¶ 546 ***TSS:***

¶ 547 The evaluator shall confirm identifiable external interfaces from guidance documents and examine that TSS description identifies all the interfaces required for realizing SFR.

¶ 548 The evaluator shall confirm identification information of the TSFI associated with the SFR described in the TSS and confirm the consistency with the description related to each interface.

¶ 549 The evaluator shall check to ensure that the SFR defined in the ST is appropriately realized, based on identification information of the TSFI in the TSS description as well as on the information of purposes, methods of use, and parameters for each TSFI in the guidance documents

¶ 550 The assurance activities specific to each SFR are described in Section 4, and also applicable SFRs from Appendix B , Appendix C , and Appendix D , and the evaluator shall perform evaluations by adding to this assurance component.

5.3 Class AGD: Guidance Documents

¶ 551 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 can fulfill its role for the security functionality. The documentation should be in an informal style and readable by an administrator.

¶ 552 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.

¶ 553 Guidance pertaining to particular security functionality is also provided; requirements on such guidance are contained in the assurance activities specified in Section 4, and applicable assurance activities in Appendix B, Appendix C, and Appendix D.

5.3.1 AGD_OPE.1 Operational user guidance

Developer action elements:

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

¶ 555 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:

¶ 556 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.

¶ 557 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.

¶ 558 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.

¶ 559 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.

¶ 560 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.

¶ 561 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.

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

Evaluator action elements:

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

¶ 564 **Assurance activity:**

¶ 565 ***Operational Guidance:***

¶ 566 The contents of operational guidance are confirmed by the assurance activities in Section 4, and applicable assurance activities in Appendix B , Appendix C , and Appendix D , and the TOE evaluation in accordance with the CEM.

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

¶ 568 Procedures for administrators to confirm that the TOE returns to its evaluation configuration after the transition from the maintenance mode to the normal Operational Environment.

¶ 569 ***Application note:***

¶ 570 *During evaluation, the TOE returns to its evaluation configuration. In the field, the TOE may return to the configuration that was in force prior to entering maintenance mode.*

5.3.2 AGD_PRE.1 Preparative procedures

Developer action elements:

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

¶ 572 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:

¶ 573 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.

¶ 574 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:

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

¶ 576 AGD_PRE.1.2E The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

¶ 577 **Assurance activity:**

¶ 578 ***Operational Guidance:***

¶ 579 The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.

5.4 Class ALC: Life-cycle Support

¶ 580 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.

5.4.1 ALC_CMC.1 Labelling of the TOE

¶ 581 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:

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

Content and presentation elements:

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

Evaluator action elements:

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

¶ 585 **Assurance activity:**

¶ 586 ***Operational Guidance:***

¶ 587 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. The evaluator shall ensure that this identifier is sufficient for an acquisition entity to use in procuring the TOE (including the appropriate administrative guidance) as specified in 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.

5.4.2 ALC_CMS.1 TOE CM coverage

¶ 588 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:

¶ 589 ALC_CMS.1.1D The developer shall provide a configuration list for the TOE.

Content and presentation elements:

¶ 590 ALC_CMS.1.1C The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

¶ 591 ALC_CMS.1.2C The configuration list shall uniquely identify the configuration items.

Evaluator action elements:

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

¶ 593 **Assurance activity:**

¶ 594 ***Operational Guidance:***

¶ 595 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.

5.5 Class ATE: Tests

¶ 596 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.

5.5.1 ATE_IND.1 Independent testing - Conformance

¶ 597 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, and applicable assurance requirements in Appendix B , Appendix C , and Appendix D are being met, although some additional testing is specified for SARs in Section 5. 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 product models combinations that are claiming conformance to this PP.

Developer action elements:

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

Content and presentation elements:

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

Evaluator action elements:

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

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

¶ 602 **Assurance activity:**

¶ 603 ***Test:***

¶ 604 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.

¶ 605 The Test Plan identifies the product models to be tested, and for those product models not included in the test plan but included in the ST, the test plan provides a justification for not testing the models. This justification must address the differences between the tested models and the untested models, 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. In case the ST describes multiple models (product names) in particular, the evaluator shall consider the differences in language specification as well as the influences, in which functions except security functions such as a printing function, may affect security functions when creating this justification. If all product models claimed in the ST are tested, then no rationale is necessary.

¶ 606 The test plan describes the composition of each product model 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 model 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.

¶ 607 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.

5.6 Class AVA: Vulnerability Assessment

¶ 608 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.

5.6.1 AVA_VAN.1 Vulnerability survey

Developer action elements:

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

Content and presentation elements:

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

Evaluator action elements:

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

evidence.

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

¶ 613 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.

¶ 614 **Assurance activity:**

¶ 615 ***Test:***

¶ 616 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 printing devices and the implemented communication protocols 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.

¶ 617 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.

¶ 618 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.

5.7 Security Assurance Requirements rationale

¶ 619 The rationale for choosing these security assurance requirements is that they define a minimum security baseline that is based on the anticipated threat level of the attacker, the

security of the Operational Environment in which the TOE is deployed, and the relative value of the TOE itself. The assurance activities throughout the PP are used to provide tailored guidance on the specific expectations for completing the security assurance requirements.

Appendix A Definitions and Rationale Tables

A.1 User Definitions

¶ 620 There are two categories of Users defined in this PP:

Table 6 User Categories

Designation	Category name	Definition
U.NORMAL	Normal User	A User who has been identified and authenticated and does not have an administrative role
U.ADMIN	Administrator	A User who has been identified and authenticated and has an administrative role

¶ 621 A conforming TOE may define additional roles, sub-roles, or groups. In particular, a conforming TOE may define several administrative roles that have authority to administer different aspects of the TOE.

A.2 Asset Definitions

¶ 622 Assets are passive entities in the TOE that contain or receive information. In this PP, Assets are Objects (as defined by the CC). There are two categories of Assets defined in this PP:

Table 7 Asset categories

Designation	Asset category	Definition
D.USER	User Data	Data created by and for Users that do not affect the operation of the TSF
D.TSF	TSF Data	Data created by and for the TOE that might affect the operation of the TSF

¶ 623 A conforming TOE may define additional Asset categories.

A.2.1 User Data

¶ 624 User Data are composed of two types:

Table 8 User Data types

Designation	User Data type	Definition
D.USER.DOC	User Document Data	Information contained in a User's Document, in electronic or hardcopy form
D.USER.JOB	User Job Data	Information related to a User's Document or Document Processing Job

¶ 625 A conforming TOE may define additional types of User Data.

A.2.2 TSF Data

¶ 626 TSF Data are composed of two types:

Table 9 TSF Data types

Designation	TSF Data type	Definition
D.TSF.PROT	Protected TSF Data	TSF Data for which alteration by a User who is neither the data owner nor in an Administrator role might affect the security of the TOE, but for which disclosure is acceptable
D.TSF.CONF	Confidential TSF Data	TSF Data for which either disclosure or alteration by a User who is neither the data owner nor in an Administrator role might affect the security of the TOE

¶ 627 A conforming TOE may define additional types of TSF Data.

A.3 Threat Definitions

¶ 628 Threats are defined by a threat agent that performs an action resulting in an outcome that has the potential to violate TOE security policies.

Table 10 Threats

Designation	Definition
T.UNAUTHORIZED_ACCESS	An attacker may access (read, modify, or delete) User Document Data or change (modify or delete) User Job Data in the TOE through one of the TOE's interfaces.
T.TSF_COMPROMISE	An attacker may gain Unauthorized Access to TSF Data in the TOE through one of the TOE's interfaces.
T.TSF_FAILURE	A malfunction of the TSF may cause loss of security if the TOE is permitted to operate.
T.UNAUTHORIZED_UPDATE	An attacker may cause the installation of unauthorized software on the TOE.
T.NET_COMPROMISE	An attacker may access data in transit or otherwise compromise the security of the TOE by monitoring or manipulating network communication.

A.4 Organizational Security Policy Definitions

¶ 629 Organizational Security Policies are used to provide a basis for Security Objectives that are not practical to define on the basis of Threats to Assets or that originate primarily from customer expectations.

Table 11 Organizational Security Policies

Designation	Definition
P.AUTHORIZATION	Users must be authorized before performing Document Processing and administrative functions.
P.AUDIT	Security-relevant activities must be audited and the log of such actions must be protected and transmitted to an External IT Entity.
P.COMMS_PROTECTION	The TOE must be able to identify itself to other devices on the LAN.
P.STORAGE_ENCRYPTION (conditionally mandatory)	If the TOE stores User Document Data or Confidential TSF Data on Field-Replaceable Nonvolatile Storage Devices, it will encrypt such data on those devices.
P.KEY_MATERIAL (conditionally mandatory)	Cleartext keys, submasks, random numbers, or any other values that contribute to the creation of encryption keys for Field-Replaceable Nonvolatile Storage of User Document Data or Confidential TSF Data must be protected from unauthorized access and must not be stored on that storage device.
P.FAX_FLOW (conditionally mandatory)	If the TOE provides a PSTN fax function, it will ensure separation between the PSTN fax line and the LAN.
P.IMAGE_OVERWRITE (optional)	Upon completion or cancellation of a Document Processing job, the TOE shall overwrite residual image data from its Field-Replaceable Nonvolatile Storage Devices.

Designation	Definition
P.PURGE_DATA (optional)	The TOE shall provide a function that an authorized administrator can invoke to make all customer-supplied User Data and TSF Data permanently irretrievable from Nonvolatile Storage Devices.

A.5 Assumption Definitions

¶ 630 Assumptions are conditions that must be satisfied in order for the Security Objectives and functional requirements to be effective.

Table 12 Assumptions

Designation	Definition
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it stores or processes, is assumed to be provided by the environment.
A.NETWORK	The Operational Environment is assumed to protect the TOE from direct, public access to its LAN interface.
A.TRUSTED_ADMIN	TOE Administrators are trusted to administer the TOE according to site security policies.
A.TRAINED_USERS	Authorized Users are trained to use the TOE according to site security policies.

A.6 Definitions of Security Objectives for the TOE

Table 13 Security Objectives for the TOE

Designation	Definition
O.USER_I&A	The TOE shall perform identification and authentication of Users for operations that require access control, User authorization, or Administrator roles.
O.ACCESS_CONTROL	The TOE shall enforce access controls to protect User Data and TSF Data in accordance with security policies.
O.USER_AUTHORIZATION	The TOE shall perform authorization of Users in accordance with security policies.
O.ADMIN_ROLES	The TOE shall ensure that only authorized Administrators are permitted to perform administrator functions.
O.UPDATE_VERIFICATION	The TOE shall provide mechanisms to verify the authenticity of software updates.
O.TSF_SELF_TEST	The TOE shall test some subset of its security functionality to help ensure that subset is operating properly.
O.COMMS_PROTECTION	The TOE shall have the capability to protect LAN communications of User Data and TSF Data from Unauthorized Access, replay, and source/destination spoofing.
O.AUDIT	The TOE shall generate audit data, and be capable of sending it to a trusted External IT Entity. Optionally, it may store audit data in the TOE.
O.STORAGE_ENCRYPTION (conditionally mandatory)	If the TOE stores User Document Data or Confidential TSF Data in Field-Replaceable Nonvolatile Storage devices, then the TOE shall encrypt such data on those devices.

Designation	Definition
O.KEY_MATERIAL (conditionally mandatory)	The TOE shall protect from unauthorized access any cleartext keys, submasks, random numbers, or other values that contribute to the creation of encryption keys for storage of User Document Data or Confidential TSF Data in Field-Replaceable Nonvolatile Storage Devices; The TOE shall ensure that such key material is not stored in cleartext on the storage device that uses that material.
O.FAX_NET_SEPARATION (conditionally mandatory)	If the TOE provides a PSTN fax function, then the TOE shall ensure separation of the PSTN fax telephone line and the LAN, by system design or active security function.
O.IMAGE_OVERWRITE (optional)	Upon completion or cancellation of a Document Processing job, the TOE shall overwrite residual image data from its Field-Replaceable Nonvolatile Storage Devices.
O.PURGE_DATA (optional)	The TOE provides a function that an authorized administrator can invoke to make all customer-supplied User Data and TSF Data permanently irretrievable from Nonvolatile Storage Devices.

A.7 Definitions of Security Objectives for the Operational Environment

Table 14 Security Objectives for the Operational Environment

Designation	Definition
OE.PHYSICAL_PROTECTION	The Operational Environment shall provide physical security, commensurate with the value of the TOE and the data it stores or processes.
OE.NETWORK_PROTECTION	The Operational Environment shall provide network security to protect the TOE from direct, public access to its LAN interface.

OE.ADMIN_TRUST	The TOE Owner shall establish trust that Administrators will not use their privileges for malicious purposes.
OE.USER_TRAINING	The TOE Owner shall ensure that Users are aware of site security policies and have the competence to follow them.
OE.ADMIN_TRAINING	The TOE Owner shall ensure that Administrators are aware of site security policies and have the competence to use manufacturer's guidance to correctly configure the TOE and protect passwords and keys accordingly.

A.8 Security Objectives Tables

Table 15 Security Objectives rationale

Threat/Policy/Assumption	Rationale
T.UNAUTHORIZED_ACCESS *An attacker may access (read, modify, or delete) User Document Data or change (modify or delete) User Job Data in the TOE through one of the TOE's interfaces.*	O.ACCESS_CONTROL restricts access to User Data in the TOE to authorized Users. O.USER_I&A provides the basis for access control. O.ADMIN_ROLES restricts the ability to authorize Users and set access controls to authorized Administrators.
T.TSF_COMPROMISE *An attacker may gain Unauthorized Access to TSF Data in the TOE through one of the TOE's interfaces.*	O.ACCESS_ CONTROL restricts access to TSF Data in the TOE to authorized Users. O.USER_I&A provides the basis for access control. O.ADMIN_ROLES restricts the ability to authorize Users and set access controls to authorized Administrators.

Threat/Policy/Assumption	Rationale
T.TSF_FAILURE *A malfunction of the TSF may cause loss of security if the TOE is permitted to operate.*	O.TSF_SELF_TEST prevents the TOE from operating if a malfunction is detected.
T.UNAUTHORIZED_UPDATE *An attacker may cause the installation of unauthorized software on the TOE.*	O.UPDATE_VERIFICATION verifies the authenticity of software updates.
T.NET_COMPROMISE *An attacker may access data in transit or otherwise compromise the security of the TOE by monitoring or manipulating network communication.*	O.COMMS_PROTECTION protects LAN communications from sniffing, replay, and man-in-the-middle attacks.
P.AUTHORIZATION *Users must be authorized before performing Document Processing and administrative functions.*	O.USER_AUTHORIZATION restricts the ability to perform Document Processing and administrative functions to authorized Users. O.USER_I&A provides the basis for authorization. O.ADMIN_ROLES restricts the ability to authorize Users to authorized Administrators.
P.AUDIT *Security-relevant activities must be audited and the log of such actions must be protected and transmitted to an External IT Entity.*	O.AUDIT requires the generation of audit data. O.ACCESS_CONTROL restricts access to audit data in the TOE to authorized Users. O.USER_AUTHORIZATION provides the basis for authorization.

Threat/Policy/Assumption	Rationale
P.COMMS_PROTECTION *The TOE* must be able to identify itself to other devices *on the LAN.*	O.COMMS_PROTECTION protects LAN communications from man-in-the-middle attacks.
P.STORAGE_ENCRYPTION (conditionally mandatory) *If the TOE stores User Document Data or Confidential TSF Data on Field-Replaceable Nonvolatile Storage Devices, it will encrypt such data on those devices.*	O.STORAGE_ENCRYPTION protects User Document Data and Confidential TSF Data stored in Field-Replaceable Nonvolatile Storage Devices from exposure if a device has been removed from the TOE and its Operational Environment.
P.KEY_MATERIAL (conditionally mandatory) *Cleartext keys, submasks, random numbers, or any other values that contribute to the creation of encryption keys for Field-Replaceable Nonvolatile Storage of User Document Data or Confidential TSF Data must be protected from unauthorized access and must not be stored on that storage device.*	O.KEY_MATERIAL protects keys and key materials from unauthorized access and ensures that they any key materials are not stored in cleartext on the device that uses those materials for its own encryption.
P.FAX_FLOW (conditionally mandatory) *If the TOE provides a PSTN fax function, it will ensure separation between the PSTN fax line and the LAN.*	O.FAX_NET_SEPARATION requires a separation between the PSTN fax line and the LAN.

Threat/Policy/Assumption	Rationale
P.IMAGE_OVERWRITE (optional) *Upon completion or cancellation of a Document Processing job, the TOE shall overwrite residual image data from its Field-Replaceable Nonvolatile Storage Device.*	O.IMAGE_OVERWRITE overwrites residual image data from Field-Replaceable Nonvolatile Storage Devices after Document Processing jobs are completed or cancelled .
P.PURGE_DATA (optional) *The TOE shall provide a function that an authorized administrator can invoke to make all customer-supplied User Data and TSF Data permanently irretrievable from Nonvolatile Storage Devices.*	O.PURGE_DATA provides a function that makes all customer-supplied User Data and TSF Data permanently irretrievable from Nonvolatile Storage Devices when invoked by an authorized administrator.
A.PHYSICAL *Physical security, commensurate with the value of the TOE and the data it stores or processes, is assumed to be provided by the environment.*	OE.PHYSICAL_PROTECTION establishes a protected physical environment for the TOE.
A.NETWORK *The Operational Environment is assumed to protect the TOE from direct, public access to its LAN interface.*	OE.NETWORK_PROTECTION establishes a protected LAN environment for the TOE.
A.TRUSTED_ADMIN *TOE Administrators are trusted to administer the TOE according to site security policies.*	OE.ADMIN_TRUST establishes responsibility of the TOE Owner to have a trusted relationship with Administrators.

Threat/Policy/Assumption	Rationale
A.TRAINED_USERS *Authorized Users are trained to use the TOE according to site security policies.*	OE.ADMIN_TRAINING establishes responsibility of the TOE Owner to provide appropriate training for Administrators. OE.USER_TRAINING establishes responsibility of the TOE Owner to provide appropriate training for Users.

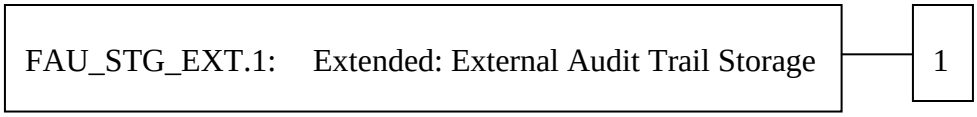
A.9 Extended Component Definitions

A.9.1 FAU_STG_EXT Extended: External Audit Trail Storage

¶ 631 **Family Behavior:**

¶ 632 This family defines requirements for the TSF to ensure that secure transmission of audit data from TOE to an External IT Entity.

¶ 633 **Component leveling:**



¶ 634 **FAU_STG_EXT.1** External Audit Trail Storage requires the TSF to use a trusted channel implementing a secure protocol.

¶ 635 **Management:**

¶ 636 The following actions could be considered for the management functions in FMT:

- The TSF shall have the ability to configure the cryptographic functionality.

¶ 637 **Audit:**

¶ 638 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 639 **FAU_STG_EXT.1 Extended: Protected Audit Trail Storage**

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation,

FTP_ITC.1 Inter-TSF trusted channel

¶ 640 **FAU_STG_EXT.1.1** The TSF shall be able to transmit the generated audit data to an External IT Entity using a trusted channel according to FTP_ITC.1.

¶ 641 **Rationale:**

¶ 642 The TSF is required that the transmission of generated audit data to an External IT Entity which relies on a non-TOE audit server for storage and review of audit records. The storage of these audit records and the ability to allow the administrator to review these audit records is provided by the Operational Environment in that case. The Common Criteria does not provide a suitable SFR for the transmission of audit data to an External IT Entity.

¶ 643 This extended component protects the audit records, and it is therefore placed in the FAU class with a single component.

A.9.2 FCS_CKM_EXT Extended: Cryptographic Key Management

¶ 644 **Family Behavior:**

¶ 645 This family addresses the management aspects of cryptographic keys. Especially, this extended component is intended for cryptographic key destruction.

¶ 646 **Component leveling:**

FCS_CKM_EXT.4: Extended: Cryptographic Key Material Destruction	4

¶ 647 **FCS_CKM_EXT.4** Cryptographic Key Material Destruction ensures not only keys but also key materials that are no longer needed are destroyed by using an approved method.

¶ 648 **Management:**

¶ 649 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 650 **Audit:**

¶ 651 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 652 **FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction**

Hierarchical to: No other components.

Dependencies: [FCS_CKM.1(a) Cryptographic Key Generation (for asymmetric keys), or

FCS_CKM.1(b) Cryptographic key generation (Symmetric Keys)],

FCS_CKM.4 Cryptographic key destruction

¶ 653 **FCS_CKM_EXT.4.1** The TSF shall destroy all plaintext secret and private cryptographic keys and cryptographic critical security parameters when no longer needed.

¶ 654 **Rationale:**

¶ 655 Cryptographic Key Material Destruction is to ensure the keys and key materials that are no longer needed are destroyed by using an approved method, and the Common Criteria does not provide a suitable SFR for the Cryptographic Key Material Destruction.

¶ 656 This extended component protects the cryptographic key and key materials against exposure, and it is therefore placed in the FCS class with a single component.

A.9.3 FCS_HTTPS_EXT Extended: HTTPS selected

¶ 657 **Family Behavior:**

¶ 658 Components in this family define requirements for protecting remote management sessions between the TOE and a Security Administrator. This family describes how HTTPS will be implemented. This is a new family defined for the FCS Class.

¶ 659 **Component leveling:**

¶ 660 **FCS_HTTPS_EXT.1** HTTPS selected, requires that HTTPS be implemented according to RFC 2818 and supports TLS.

¶ 661 **Management:**

¶ 662 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 663 **Audit:**

¶ 664 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- Failure of HTTPS session establishment

¶ 665 **FCS_HTTPS_EXT.1Extended: HTTPS selected**

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 666 **FCS_HTTPS_EXT.1.1** The TSF shall implement the HTTPS protocol that complies with RFC 2818.

¶ 667 **FCS_HTTPS_EXT.1.2** The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

¶ 668 **Rationale:**

¶ 669 HTTPS is one of the secure communication protocols, and the Common Criteria does not provide a suitable SFR for the communication protocols using cryptographic algorithms.

¶ 670 This extended component protects the communication data using cryptographic algorithms, and it is therefore placed in the FCS class with a single component.

A.9.4 FCS_IPSEC_EXT Extended: IPsec selected

¶ 671 **Family Behavior:**

¶ 672 This family addresses requirements for protecting communications using IPsec.

¶ 673 **Component leveling:**

¶ 674 **FCS_IPSEC_EXT.1** IPsec requires that IPsec be implemented as specified.

¶ 675 **Management:**

¶ 676 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 677 **Audit:**

¶ 678 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- Failure to establish an IPsec SA

¶ 679 **FCS_IPSEC_EXT.1 Extended: IPsec selected**

Hierarchical to: No other components.

Dependencies: FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition

FCS_COP.1(g) Cryptographic Operation (for keyed-hash message authentication)

¶ 680 **FCS_IPSEC_EXT.1.1** The TSF shall implement the IPsec architecture as specified in RFC 4301.

¶ 681 **FCS_IPSEC_EXT.1.2** The TSF shall implement [selection: tunnel mode, transport mode].

¶ 682 **FCS_IPSEC_EXT.1.3** The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

¶ 683 **FCS_IPSEC_EXT.1.4** The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using [selection: *the cryptographic algorithms AES-CBC-128 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, AES-CBC-256 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, AES-GCM-128 as specified in RFC 4106, AES-GCM-256 as specified in RFC 4106*].

¶ 684 **FCS_IPSEC_EXT.1.5** The TSF shall implement the protocol: [selection: *IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109,* [selection: *no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers*], *and* [selection: *no other RFCs for hash functions, RFC 4868 for hash functions*]; *IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and* [selection: *no other RFCs for hash functions, RFC 4868 for hash functions*]].

¶ 685 **FCS_IPSEC_EXT.1.6** The TSF shall ensure the encrypted payload in the [selection: *IKEv1, IKEv2*] protocol uses the cryptographic algorithms AES-CBC-128,

AES-CBC-256 as specified in RFC 3602 and [selection: *AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm*].

¶ 686 **FCS_IPSEC_EXT.1.7** The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

¶ 687 **FCS_IPSEC_EXT.1.8** The TSF shall ensure that [selection: *IKEv2 SA lifetimes can be established based on [*selection*: number of packets/number of bytes; length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs]; IKEv1 SA lifetimes can be established based on [*selection*: number of packets/number of bytes ; length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs*]].

¶ 688 **FCS_IPSEC_EXT.1.9** The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [selection: *24 (2048-bit MODP with 256-bit POS), 19 (256-bit Random ECP), 20 (384-bit Random ECP, 5 (1536-bit MODP))*, [assignment: *other DH groups that are implemented by the TOE], no other DH groups*].

¶ 689 **FCS_IPSEC_EXT.1.10** The TSF shall ensure that all IKE protocols perform Peer Authentication using the [selection: *RSA, ECDSA*] algorithm and Pre-shared Keys.

¶ 690 **Rationale:**

¶ 691 IPsec is one of the secure communication protocols, and the Common Criteria does not provide a suitable SFR for the communication protocols using cryptographic algorithms.

¶ 692 This extended component protects the communication data using cryptographic algorithms, and it is therefore placed in the FCS class with a single component.

A.9.5 FCS_KDF_EXT Extended: Cryptographic Key Derivation

¶ 693 **Family Behavior**

¶ 694 This family specifies the means by which an intermediate key is derived from a specified set of submasks.

¶ 695 **Component leveling**

FCS_KDF_EXT Cryptographic Key Derivation	1

¶ 696 **FCS_KDF_EXT.1** Cryptographic Key Derivation requires the TSF to derive

intermediate keys from submasks using the specified hash functions.

¶ 697 **Management:**

¶ 698 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 699 **Audit:**

¶ 700 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 701 **FCS_KDF_EXT.1** Extended: Cryptographic Key Derivation

Hierarchical to: No other components

Dependencies: FCS_COP.1(h) Cryptographic Operation (for keyed-hash message authentication),

[if selected: FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)]

¶ 702 **FCS_KDF_EXT.1.1** The TSF shall accept [selection: *a RNG generated submask as specified in FCS_RBG_EXT.1, a conditioned password submask, imported submask*] to derive an intermediate key, as defined in [selection: *NIST SP 800-108* [selection: *KDF in Counter Mode, KDF in Feedback Mode, KDF in Double-Pipeline Iteration Mode]*, *NIST SP 800-132*], using the keyed-hash functions specified in FCS_COP.1(h), such that the output is at least of equivalent security strength (in number of bits) to the BEV.

A.9.6 FCS_KYC_EXT Extended: Cryptographic Operation (Key Chaining)

¶ 703 **Family Behavior:**

¶ 704 This family provides the specification to be used for using multiple layers of encryption keys to ultimately secure the protected data encrypted on the storage.

¶ 705 **Component leveling:**

¶ 706 **FCS_KYC_EXT** Key Chaining, requires the TSF to maintain a key chain and specifies

the characteristics of that chain.

¶ 707 **Management:**

¶ 708 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 709 **Audit:**

¶ 710 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 711 **FCS_KYC_EXT.1 Extended: Key Chaining**

Hierarchical to: No other components.

Dependencies: [FCS_COP.1(e) Cryptographic operation (Key Wrapping),
FCS_SMC_EXT.1 Extended: Submask Combining,
FCS_COP.1(i) Cryptographic operation (Key Transport),
FCS_KDF_EXT.1 Cryptographic Operation (Key Derivation),
and/or
FCS_COP.1(f) Cryptographic operation (Key Encryption)].

¶ 712 **FCS_KYC_EXT.1.1** The TSF shall maintain a key chain of: [selection: *one, using a submask as the BEVor DEK*; *intermediate keys originating from one or more submask(s) to the BEV or DEK using the following method(s)*: [selection: *key wrapping as specified in FCS_COP.1(e), key combining as specified in FCS_SMC_EXT.1, key encryption as specified in FCS_COP.1(f), key derivation as specified in FCS_KDF_EXT.1, key transport as specified in FCS_COP.1(i)*]] while maintaining an effective strength of [selection: *128 bits, 256 bits*].

¶ 713 **Rationale:**

¶ 714 Key Chaining ensures that the TSF maintains the key chain, and also specifies the characteristics of that chain. However, the Common Criteria does not provide a suitable SFR for the management of multiple layers of encryption key to protect encrypted data.

¶ 715 This extended component protects the TSF data using cryptographic algorithms, and it is therefore placed in the FCS class with a single component.

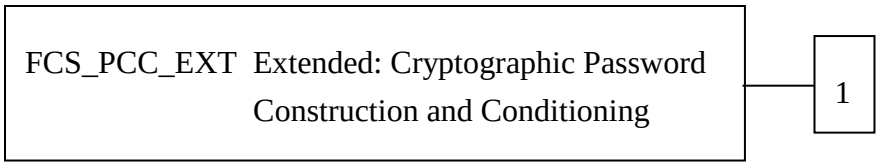
A.9.7 FCS_PCC_EXT Extended: Cryptographic Password Construction and Conditioning

¶ 716 **Family Behavior**

¶ 717 This family ensures that passwords used to produce the BEV are robust (in terms of their composition) and are conditioned to provide an appropriate-length bit string.

¶ 718 **Component leveling**

¶ 719



¶ 720 **FCS_PCC_EXT.1** Cryptographic Password Construction and Conditioning, requires the TSF to accept passwords of a certain composition and condition them appropriately.

¶ 721 **Management:**

¶ 722 No specific management functions are identified

¶ 723 **Audit:**

¶ 724 There are no auditable events foreseen.

¶ 725 **FCS_PCC_EXT.1 Extended: Cryptographic Password Construction and Conditioning**

Hierarchical to: No other components

Dependencies: FCS_COP.1(h) Cryptographic Operation (for keyed-hash message authentication)

¶ 726 **FCS_PCC_EXT.1.1** A password used to generate a password authorization factor shall enable up to [assignment: *positive integer of 64 or more*] characters in the set of {upper case characters, lower case characters, numbers, and [assignment: *other supported special characters*]} and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm [*HMAC*-[selection: *SHA-256, SHA-384, SHA-512*]], with [assignment: *positive integer of 1000 or more*] iterations, and output cryptographic key sizes [selection: *128, 256*] that meet the following: [assignment: *PBKDF recommendation or specification*].

A.9.8 FCS_RBG_EXT Extended: Cryptographic Operation (Random Bit Generation)

¶ 727 **Family Behavior:**

¶ 728 This family defines requirements for random bit generation to ensure that it is performed in accordance with selected standards and seeded by an entropy source.

¶ 729 **Component leveling:**

¶ 730 **FCS_RBG_EXT.1** Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

¶ 731 **Management:**

¶ 732 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 733 **Audit:**

¶ 734 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 735 **FCS_RBG_EXT.1 Extended: Random Bit Generation**

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 736 **FCS_RBG_EXT.1.1** The TSF shall perform all deterministic random bit generation services in accordance with [selection: *ISO/IEC 18031:2011, NIST SP 800-90A*] using [selection: *Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)*].

¶ 737 **FCS_RBG_EXT.1.2** The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [selection: [assignment: *number of software-based sources*] *software-based noise source(s)*, [assignment: *number of hardware-based sources*] *hardware-based noise source(s)*] with a minimum of [selection: *128 bits, 256 bits*] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011

Table C.1 “Security strength table for hash functions”, of the keys and hashes that it will generate.

¶ 738 **Rationale:**

¶ 739 Random bits/number will be used by the SFRs for key generation and destruction, and the Common Criteria does not provide a suitable SFR for the random bit generation.

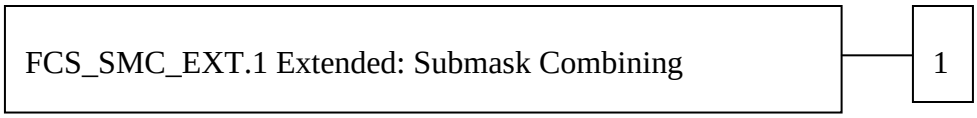
¶ 740 This extended component ensures the strength of encryption keys, and it is therefore placed in the FCS class with a single component.

A.9.9 FCS_SMC_EXT Extended: Submask Combining

¶ 741 **Family Behavior:**

¶ 742 This family defines the means by which submasks are combined, if the TOE supports more than one submask being used to derive or protect the BEV.

¶ 743 **Component leveling:**



¶ 744 **FCS_SMC_EXT.1** Submask combining requires the TSF to combine the submasks in a predictable fashion.

¶ 745 **Management:**

¶ 746 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 747 **Audit:**

¶ 748 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 749 **FCS_SMC_EXT.1 Extended: Submask Combining**

Hierarchical to: No other components.

Dependencies: FCS_COP.1(c) Cryptographic operation (Hash Algorithm)

¶ 750 **FCS_SMC_EXT.1.1** The TSF shall combine submasks using the following method

[selection: *exclusive OR (XOR), SHA-256, SHA-512*] to generate an intermediary key or BEV.

¶ 751 **Rationale:**

¶ 752 Submask Combining is to ensure the TSF combine the submasks in order to derive or protect the BEV.

¶ 753 This extended component protects the TSF data using cryptographic algorithms, and it is therefore placed in the FCS class with a single component.

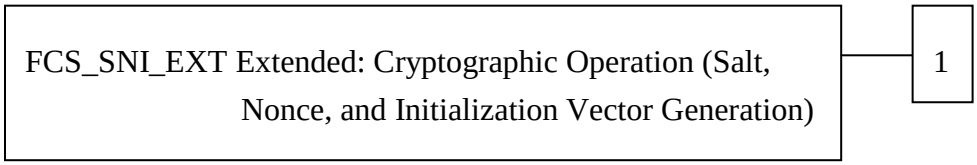
A.9.10 FCS_SNI_EXT Extended: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation

¶ 754 **Family Behavior**

¶ 755 This family ensures that salts, nonces, and IVs are well formed.

¶ 756 **Component leveling**

¶ 757



¶ 758 **FCS_SNI_EXT.1** Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation), requires the generation of salts, nonces, and IVs to be used by the cryptographic components of the TOE to be performed in the specified manner.

¶ 759 **Management:**

¶ 760 No specific management functions are identified

¶ 761 **Audit:**

¶ 762 There are no auditable events foreseen.

¶ 763 **FCS_SNI_EXT.1 Extended: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)**

Hierarchical to: No other components

Dependencies: FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

¶ 764 **FCS_SNI_EXT.1.1** The TSF shall only use salts that are generated by a **RNG as**

specified in FCS_RBG_EXT.1

¶ 765 **FCS_SNI_EXT.1.2** The TSF shall only use unique nonces with a minimum size of [64] bits.

¶ 766 **FCS_SNI_EXT.1.3** The TSF shall create IVs in the following manner: [

¶ 767 CBC: IVs shall be non-repeating,

¶ 768 CCM: Nonce shall be non-repeating.

¶ 769 XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer,

¶ 770 GCM: IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key.

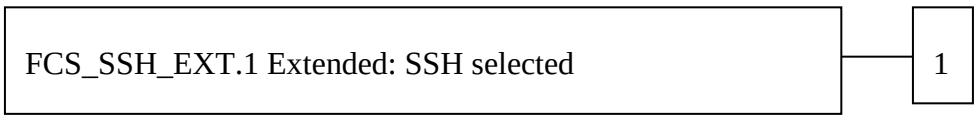
¶ 771].

A.9.11 FCS_SSH_EXT Extended: SSH selected

¶ 772 **Family Behavior:**

¶ 773 This family addresses the ability for a server and/or a client to offer SSH to protect data between a client and the server using the SSH protocol.

¶ 774 **Component leveling:**



¶ 775 **FCS_SSH_EXT.1** SSH selected, requires the SSH protocol implemented as specified.

¶ 776 **Management:**

¶ 777 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 778 **Audit:**

¶ 779 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- Failure of SSH session establishment

¶ 780 **FCS_SSH_EXT.1 Extended: SSH selected**

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 781 **FCS_SSH_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: *5656, 6668, no other RFCs*].

¶ 782 **FCS_SSH_EXT.1.2** The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

¶ 783 **FCS_SSH_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: *number of bytes*] bytes in an SSH transport connection are dropped.

¶ 784 **FCS_SSH_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [selection: *AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other algorithms*].

¶ 785 **FCS_SSH_EXT.1.5** The TSF shall ensure that the SSH transport implementation uses [selection: *SSH_RSA, ecdsa-sha2-nistp256*] and [selection: *PGP-SIGN-RSA, PGP-SIGN-DSS, ecdsa-sha2-nistp384, no other public key algorithms*,] as its public key algorithm(s).

¶ 786 **FCS_SSH_EXT.1.6** The TSF shall ensure that data integrity algorithms used in SSH transport connection is [selection: *HMAC-SHA1, HMAC-SHA1-96, HMAC-SHA2-256, HMAC-SHA2-512*].

¶ 787 **FCS_SSH_EXT.1.7** The TSF shall ensure that diffie-hellman-group14-sha1 and [selection: *ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods*] are the only allowed key exchange method used for the SSH protocol.

¶ 788 **Rationale:**

¶ 789 SSH is one of the secure communication protocols, and the Common Criteria does not provide a suitable SFR for the communication protocols using cryptographic algorithms.

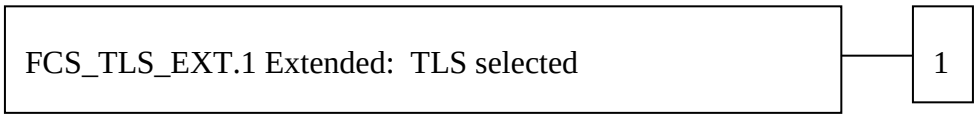
¶ 790 This extended component protects the communication data using cryptographic algorithms, and it is therefore placed in the FCS class with a single component.

A.9.12 FCS_TLS_EXT Extended: TLS selected

¶ 791 **Family Behavior:**

¶ 792 This family addresses the ability for a server and/or a client to use TLS to protect data between a client and the server using the TLS protocol.

¶ 793 **Component leveling:**



¶ 794 **FCS_TLS_EXT.1** TLS selected, requires the TLS protocol implemented as specified.

¶ 795 **Management:**

¶ 796 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 797 **Audit:**

¶ 798 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- Failure of TLS session establishment

¶ 799 **FCS_TLS_EXT.1 Extended: TLS selected**

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 800 **FCS_TLS_EXT.1.1** The TSF shall implement one or more of the following protocols [selection: *TLS 1.0 (RFC 2246)*, *TLS 1.1 (RFC 4346)*, *TLS 1.2 (RFC 5246)*] supporting the following ciphersuites:

¶ 801 Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA

¶ 802 Optional Ciphersuites:

¶ 803 [selection:

- *None*
- *TLS_RSA_WITH_AES_256_CBC_SHA*
- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA*
- *TLS_RSA_WITH_AES_128_CBC_SHA256*
- *TLS_RSA_WITH_AES_256_CBC_ SHA256*
- *TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256*
- *TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384*

].

¶ 804 **Rationale:**

¶ 805 TLS is one of the secure communication protocols, and the Common Criteria does not provide a suitable SFR for the communication protocols using cryptographic algorithms.

¶ 806 This extended component protects the communication data using cryptographic

algorithms, and it is therefore placed in the FCS class with a single component.

A.9.13 FDP_DSK_EXT Extended: Protection of Data on Disk

¶ 807 **Family Behavior:**

¶ 808 This family is to mandate the encryption of all protected data written to the storage.

¶ 809 **Component leveling:**

¶ 810 **FDP_DSK_EXT.1** Extended: Protection of Data on Disk, requires the TSF to encrypt all the Confidential TSF and User Data stored on the Field-Replaceable Nonvolatile Storage Devices in order to avoid storing these data in plaintext on the devices.

¶ 811 **Management:**

¶ 812 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 813 **Audit:**

¶ 814 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 815 **FDP_DSK_EXT.1 Extended: Protection of Data on Disk**

Hierarchical to: No other components.

Dependencies: FCS_COP.1(d) Cryptographic operation (AES Data Encryption/Decryption)

¶ 816 **FDP_DSK_EXT.1.1** The TSF shall [selection: *perform encryption in accordance with FCS_COP.1(d), use a self-encrypting Field-Replaceable Nonvolatile Storage Device that is separately CC certified to conform to the FDE EE cPP*] such that any Field-Replaceable Nonvolatile Storage Device contains no plaintext User Document Data and no plaintext confidential TSF Data.

¶ 817 **FDP_DSK_EXT.1.2** The TSF shall encrypt all protected data without user intervention.

¶ 818 **Rationale:**

¶ 819 Extended: Protection of Data on Disk is to specify that encryption of any confidential data without user intervention, and the Common Criteria does not provide a suitable SFR for the Protection of Data on Disk.

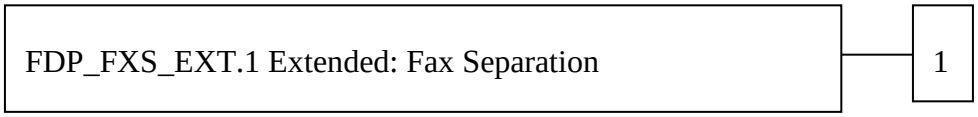
¶ 820 This extended component protects the Data on Disk, and it is therefore placed in the FDP class with a single component.

A.9.14 FDP_FXS_EXT Extended: Fax Separation

¶ 821 **Family Behavior:**

¶ 822 This family addresses the requirements for separation between Fax PSTN line and the LAN to which TOE is connected.

¶ 823 **Component leveling:**



¶ 824 **FDP_FXS_EXT.1** Fax Separation, requires the fax interface cannot be used to create a network bridge between a PSTN and a LAN to which TOE is connected.

¶ 825 **Management:**

¶ 826 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 827 **Audit:**

¶ 828 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 829 **FDP_FXS_EXT.1 Extended: Fax separation**

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 830 **FDP_FXS_EXT.1.1** The TSF shall prohibit communication via the fax interface, except transmitting or receiving User Data using fax protocols.

¶ 831 **Rationale:**

¶ 832 Fax Separation is to protect a LAN against attack from PSTN line, and the Common Criteria does not provide a suitable SFR for the Protection of TSF or User Data.

¶ 833 This extended component protects the TSF Data or User Data, and it is therefore placed in the FDP class with a single component.

A.9.15 FIA_PMG_EXT Extended: Password Management

¶ 834 **Family Behavior:**

¶ 835 This family defines requirements for the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

¶ 836 **Component leveling:**

¶ 837 **FIA_PMG _EXT.1** Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

¶ 838 **Management:**

¶ 839 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 840 **Audit:**

¶ 841 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 842 **FIA_PMG _EXT.1 Extended: Password management**

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 843 **FIA_PMG _EXT.1.1** The TSF shall provide the following password management capabilities for User passwords:

- Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [selection: *"!"*, *"@"*, *"#"*, *"$"*, *"%"*, "^", *"&"*, "*", *"("*, *")"*, [assignment: *other characters*]];
- Minimum password length shall be settable by an Administrator, and have the capability to require passwords of 15 characters or greater.

¶ 844 **Rationale:**

¶ 845 Password Management is to ensure the strong authentication between the endpoints of communication, and the Common Criteria does not provide a suitable SFR for the Password Management.

¶ 846 This extended component protects the TOE by means of password management, and it is therefore placed in the FIA class with a single component.

A.9.16 FIA_PSK_EXT Extended: Pre-Shared Key Composition

¶ 847 **Family Behavior:**

¶ 848 This family defines requirements for the TSF to ensure the ability to use pre-shared keys for IPsec.

¶ 849 **Component leveling:**

¶ 850 **FIA_PSK_EXT.1** Pre-Shared Key Composition, ensures authenticity and access control for updates.

¶ 851 **Management:**

¶ 852 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 853 **Audit:**

¶ 854 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 855 **FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition**

Hierarchical to: No other components.

Dependencies: FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation).

¶ 856 **FIA_PSK_EXT.1.1** The TSF shall be able to use pre-shared keys for IPsec.

¶ 857 **FIA_PSK_EXT.1.2** The TSF shall be able to accept text-based pre-shared keys that are:

- 22 characters in length and [selection: [assignment: *other supported lengths], no other lengths*];
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")").

¶ 858 **FIA_PSK_EXT.1.3** The TSF shall condition the text-based pre-shared keys by using [selection: *SHA-1, SHA-256, SHA-512*, [assignment: *method of conditioning text string*]] and be able to [selection: *use no other pre-shared keys; accept bit-based pre-shared keys; generate bit-based pre-shared keys using the random bit generator specified in FCS_RBG_EXT.1*].

¶ 859 **Rationale:**

¶ 860 Pre-shared Key Composition is to ensure the strong authentication between the endpoints of communications, and the Common Criteria does not provide a suitable SFR for the Pre-shared Key Composition.

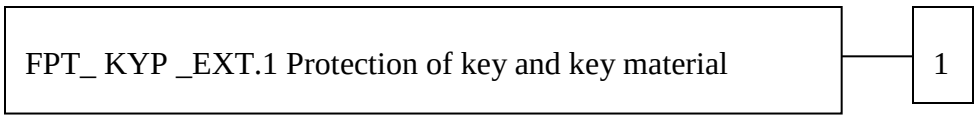
¶ 861 This extended component protects the TOE by means of strong authentication, and it is therefore placed in the FIA class with a single component.

A.9.17 FPT_KYP_EXT Extended: Protection of Key and Key Material

¶ 862 **Family Behavior:**

¶ 863 This family addresses the requirements for keys and key materials to be protected if and when written to nonvolatile storage.

¶ 864 **Component leveling:**



¶ 865

¶ 866 **FPT_ KYP _EXT.1** Extended: Protection of key and key material, requires the TSF to ensure that no plaintext key or key materials are written to nonvolatile storage.

¶ 867 **Management:**

¶ 868 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 869 **Audit:**

¶ 870 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 871 **FPT_ KYP _EXT.1 Extended: Protection of Key and Key Material**

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 872 **FPT_ KYP _EXT.1.1** The TSF shall not store plaintext keys that are part of the keychain specified by FCS_KYC_EXT.1 in any Field-Replaceable Nonvolatile Storage Device, and not store any such plaintext key on a device that uses the key for its encryption.

¶ 873 **Rationale:**

¶ 874 Protection of Key and Key Material is to ensure that no plaintext key or key material are written to nonvolatile storage, and the Common Criteria does not provide a suitable SFR for the protection of key and key material.

¶ 875 This extended component protects the TSF data, and it is therefore placed in the FPT class with a single component.

A.9.18 FPT_SKP_EXT Extended: Protection of TSF Data

¶ 876 **Family Behavior:**

¶ 877 This family addresses the requirements for managing and protecting the TSF data, such as cryptographic keys. This is a new family modelled as the FPT Class.

¶ 878 **Component leveling:**

¶ 879 **FPT_SKP_EXT.1** Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

¶ 880 **Management:**

¶ 881 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 882 **Audit:**

¶ 883 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 884 **FPT_SKP_EXT.1 Extended: Protection of TSF Data**

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 885 **FPT_SKP_EXT.1.1** The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

¶ 886 **Rationale:**

¶ 887 Protection of TSF Data is to ensure the pre-shared keys, symmetric keys and private keys are protected securely, and the Common Criteria does not provide a suitable SFR for the protection of such TSF data.

¶ 888 This extended component protects the TOE by means of strong authentication using Pre-shared Key, and it is therefore placed in the FPT class with a single component.

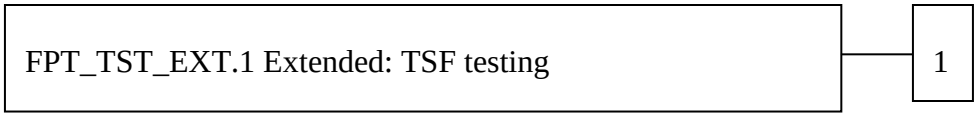
A.9.19 FPT_TST_EXT Extended: TSF testing

¶ 889 **Family Behavior:**

¶ 890 This family addresses the requirements for self-testing the TSF for selected correct

operation.

¶ 891 **Component leveling:**



¶ 892 **FPT_TST_EXT.1** TSF testing requires a suite of self-testing to be run during initial start-up in order to demonstrate correct operation of the TSF.

¶ 893 **Management:**

¶ 894 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 895 **Audit:**

¶ 896 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 897 **FPT_TST_EXT.1 Extended: TSF testing**

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 898 **FPT_TST_EXT.1.1** The TSF shall run a suite of self-tests during initial start-up (and power on) to demonstrate the correct operation of the TSF.

¶ 899 **Rationale:**

¶ 900 TSF testing is to ensure the TSF can be operated correctly, and the Common Criteria does not provide a suitable SFR for the TSF testing. In particular, there is no SFR defined for TSF testing.

¶ 901 This extended component protects the TOE, and it is therefore placed in the FPT class with a single component.

A.9.20 FPT_TUD_EXT Extended: Trusted Update

¶ 902 **Family Behavior:**

¶ 903 This family defines requirements for the TSF to ensure that only administrators can

update the TOE firmware/software, and that such firmware/software is authentic.

¶ 904 **Component leveling:**

¶ 905 **FPT_TUD_EXT.1** Trusted Update, ensures authenticity and access control for updates.

¶ 906 **Management:**

¶ 907 The following actions could be considered for the management functions in FMT:

- There are no management actions foreseen.

¶ 908 **Audit:**

¶ 909 The following actions should be auditable if FAU_GEN Security Audit Data Generation is included in the PP/ST:

- There are no auditable events foreseen.

¶ 910 **FPT_TUD_EXT.1** Trusted Update

Hierarchical to: No other components.

Dependencies: [FCS_COP.1(b) Cryptographic Operation (for signature generation/verification), or

FCS_COP.1(c) Cryptographic operation (Hash Algorithm)].

¶ 911 **FPT_TUD_EXT.1.1** The TSF shall provide authorized administrators the ability to query the current version of the TOE firmware/software.

¶ 912 **FPT_TUD_EXT.1.2** The TSF shall provide authorized administrators the ability to initiate updates to TOE firmware/software.

¶ 913 **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.

¶ 914 **Rationale:**

¶ 915 Firmware/software is a form of TSF Data, and the Common Criteria does not provide a suitable SFR for the management of firmware/software. In particular, there is no SFR

defined for importing TSF Data.

¶ 916 This extended component protects the TOE, and it is therefore placed in the FPT class with a single component.

A.10 Security Functional Requirements Tables

Table 16 Security Functional Requirements completeness

¶ 917 ***Legend:***

¶ 918 **R = Required**

¶ 919 **C = Conditionally Mandatory**

¶ 920 **O = Optional**

¶ 921 **S = Selection**

¶ 922 **U = an SFR that plays a supporting role to other SFRs**

Objective: / **SFR:**	**O.ACCESS_CONTROL**	**O.ADMIN_ROLES**	**O.AUDIT**	**O.COMMS_PROTECTION**	**O.FAX_NET_SEPARATION**	**O.IMAGE_OVERWRITE**	**O.KEY_MATERIAL**	**O.PURGE_DATA**	**O.STORAGE_ENCRYPTION**	**O.TSF_SELF_TEST**	**O.UPDATE_VERIFICATION**	**O.USER_AUTHORIZATION**	**O.USER_I&A**
FAU_GEN.1			R										
FAU_GEN.2			R										
FAU_SAR.1			O										
FAU_SAR.2			O										
FAU_STG.1			O										
FAU_STG.4			O										
FAU_STG_EXT.1			R										
FCS_CKM.1(a)				R									
FCS_CKM.1(b)				R					S				
FCS_CKM.4				U				O	U				
FCS_CKM_EXT.4				U				O	U				
FCS_COP.1(a)				R									

Objective: / SFR:	O.ACCESS_CONTROL	O.ADMIN_ROLES	O.AUDIT	O.COMMS_PROTECTION	O.FAX_NET_SEPARATION	O.IMAGE_OVERWRITE	O.KEY_MATERIAL	O.PURGE_DATA	O.STORAGE_ENCRYPTION	O.TSF_SELF_TEST	O.UPDATE_VERIFICATION	O.USER_AUTHORIZATION	O.USER_I&A
FCS_COP.1(b)				S							S		
FCS_COP.1(c)									U		S		
FCS_COP.1(d)									U				
FCS_COP.1(e)									U				
FCS_COP.1(f)									U				
FCS_COP.1(g)				S									
FCS_COP.1(h)									O				
FCS_COP.1(i)									U				
FCS_HTTPS_EXT.1				S									
FCS_IPSEC_EXT.1				S									
FCS_KDF_EXT.1									O				
FCS_KYC_EXT.1									C				
FCS_PCC_EXT.1									O				
FCS_RBG_EXT.1				U					U				
FCS_SMC_EXT.1									S				
FCS_SNI_EXT.1									S				
FCS_SSH_EXT.1				S									
FCS_TLS_EXT.1				S									
FDP_ACC.1	R											R	
FDP_ACF.1	R											R	
FDP_DSK_EXT.1									C				
FDP_FXS_EXT.1					C								
FDP_RIP.1(a)						O							
FDP_RIP.1(b)								O					
FIA_AFL.1													U
FIA_ATD.1												U	
FIA_PMG_EXT.1													R
FIA_PSK_EXT.1				S									
FIA_UAU.1													R

Objective: / SFR:	O.ACCESS_CONTROL	O.ADMIN_ROLES	O.AUDIT	O.COMMS_PROTECTION	O.FAX_NET_SEPARATION	O.IMAGE_OVERWRITE	O.KEY_MATERIAL	O.PURGE_DATA	O.STORAGE_ENCRYPTION	O.TSF_SELF_TEST	O.UPDATE_VERIFICATION	O.USER_AUTHORIZATION	O.USER_I&A
FIA_UAU.7													R
FIA_UID.1		U											R
FIA_USB.1													R
FMT_MOF.1		R											
FMT_MSA.1	U											R	
FMT_MSA.3	U											R	
FMT_MTD.1	U												
FMT_SMF.1	U	R										R	
FMT_SMR.1	U	R										R	
FPT_KYP_EXT.1							C						
FPT_SKP_EXT.1				R									
FPT_STM.1			U										
FPT_TST_EXT.1										R			
FPT_TUD_EXT.1											R		
FTA_SSL.3													R
FTP_ITC.1			U	R									
FTP_TRP.1(a)				R									
FTP_TRP.1(b)				R									

Table 17 Security Functional Requirements rationale

Objective / SFR	Relationship	Rationale
O.ACCESS_CONTROL - ***The TOE shall enforce access controls to protect User Data and TSF Data in accordance with security policies***.		
FDP_ACC.1	Satisfies	This SFR defines the access control policy that is used to protect access to User Data and TSF Data.
FDP_ACF.1	Satisfies	This SFR defines the specific rule-set that constitutes the access control policy, identifying the conditions under which access to resources, functions, and data are authorized or denied."
FMT_MSA.1	Supports	The management of the product configuration, security settings, and user attributes and authorizations is critical to maintaining operational security. These management functions, as a group, provide for the ability of authorized administrators to configure the system, add and delete users, grant user-specific authorizations to system data, resources, and functions, introduce code (e.g., updates) into the system, and assign users to roles. Additionally, the SFRs also require that management functions be limited to users who have been explicitly authorized to perform management functions.
FMT_MSA.3	Supports	
FMT_MTD.1	Supports	
FMT_SMF.1	Supports	
FMT_SMR.1	Supports	
O.ADMIN_ROLES - ***The TOE shall ensure that only authorized Administrators are permitted to perform administrator functions***.		
FIA_UID.1	Supports	This SFR defines the TOE management functions that can be accessed without requiring Administrator authorization.
FMT_MOF.1	Satisfies	This SFR defines the authorizations that are required for Administrators to access TOE functions.
FMT_SMF.1	Satisfies	This SFR defines the administrative functions that are provided by the TSF.
FMT_SMR.1	Satisfies	This SFR defines the different roles that can be assigned to Administrators for the purposes of determining authentication and authorization.
O.COMMS_PROTECTION - ***The TOE shall have the capability to protect LAN communications of User Data and TSF Data from Unauthorized Access, replay, and source/destination spoofing***.		
FCS_CKM.1(a)	Satisfies	This SFR defines the use of secure algorithms for key pair generation that can be used for key transport during protected communications.
FCS_CKM.1(b)	Satisfies	This SFR defines the use of secure algorithms for key generation that can be used for protection communications.

Objective / SFR	Relationship	Rationale
FCS_CKM.4	Supports	This SFR defines the method of data erasure used by FCS_CKM_EXT.4 that provides assurance that cryptographic keys that need to be erased cannot be recovered.
FCS_CKM_EXT.4	Supports	This SFR ensures that residual cryptographic data cannot be used to compromise protected communications.
FCS_COP.1(a)	Satisfies	This SFR defines the use of a secure symmetric key algorithm that can be used for protected communications.
FCS_COP.1(g)	Selection	This SFR defines the use of a secure HMAC algorithm that can be used for protected communications.
FCS_HTTPS_EXT.1	Selection	These SFRs define secure communications protocols that can be used to protect the transmission of security-relevant data.
FCS_IPSEC_EXT.1	Selection	
FCS_RBG_EXT.1	Supports	This SFR supports protected communications by defining a secure method of random bit generation that allows cryptographic functions to operate with their theoretical maximum strengths.
FCS_SSH_EXT.1	Selection	These SFRs define secure communications protocols that can be used to protect the transmission of security-relevant data.
FCS_TLS_EXT.1	Selection	
FIA_PSK_EXT.1	Selection	This SFR defines the use of pre-shared keys in IPsec which allows for the secure implementation of that protocol.
FPT_SKP_EXT.1	Satisfies	This SFR prevents the compromise of protected communications by ensuring that secret cryptographic data is protected against unauthorized access.
FTP_ITC.1	Satisfies	This SFR defines the interfaces over which protected communications are required and the methods used to protect the communications used to transit those interfaces.
FTP_TRP.1(a)	Satisfies	This SFR defines the protected communications path that is used to secure Administrator interaction with the TOE.
FTP_TRP.1(b)	Satisfies	This SFR defines the protected communications path that is used to secure user interaction with the TOE.
O.FAX_NET_SEPARATION (Conditionally Mandatory) - *If the TOE provides a PSTN fax function, then the TOE shall ensure separation of the PSTN fax telephone line and the LAN, by system design or active security function*.		
FDP_FXS_EXT.1	Satisfies	This SFR enforces separation of the fax interface by preventing the use of this interface for all non-fax communications.
O.IMAGE_OVERWRITE (Optional) - *Upon completion or cancellation of a Document Processing job, the TOE shall overwrite residual image data from its Field-Replaceable Nonvolatile Storage Devices*.		

Objective / SFR	Relationship	Rationale
FDP_RIP.1(a)	Satisfies	This SFR defines the ability of the TSF to overwrite user document data upon its deallocation.
O.KEY_MATERIAL (Conditionally Mandatory) - ***The TOE shall protect from unauthorized access any cleartext keys, submasks, random numbers, or other values that contribute to the creation of encryption keys for storage of User Document Data or Confidential TSF Data in Field-Replaceable Nonvolatile Storage Devices; The TOE shall ensure that such key material is not stored in cleartext on the storage device that uses that material*.**		
FPT_KYP_EXT.1	Satisfies	This SFR defines the ability of the TSF from storing unprotected key data in insecure locations.
O.PURGE_DATA (Optional) - ***The TOE provides a function that an authorized administrator can invoke to make all customer-supplied User Data and TSF Data permanently irretrievable from Nonvolatile Storage Devices*.**		
FCS_CKM.4	Satisfies	This SFR defines the physical mechanism used to accomplish the data purge defined by FCS_CKM_EXT.4.
FCS_CKM_EXT.4	Satisfies	This SFR defines the ability of the TSF to purge data from storage.
FDP_RIP.1(b)	Satisfies	This SFR requires the TSF to purge all User Data and TSF Data as part of the decommissioning process.
O.STORAGE_ENCRYPTION (Conditionally Mandatory) - ***If the TOE stores User Document Data or Confidential TSF Data in Field-Replaceable Nonvolatile Storage devices, then the TOE shall encrypt such data on those devices*.**		
FCS_CKM.1(b)	Selection	This SFR defines the use of secure algorithms for key generation that can be used for storage encryption.
FCS_CKM.4	Supports	This SFR helps define the requirements for the proper destruction of cryptographic keys in order to ensure that stored data is unrecoverable should the storage device(s) be separated from the TOE.
FCS_CKM_EXT.4	Supports	This SFR helps define the requirements for the proper destruction of cryptographic keys in order to ensure that stored data is unrecoverable should the storage device(s) be separated from the TOE.
FCS_COP.1(c)	Supports	This SFR provides the ability to generate strong encryption keys using a shorter string for input in order to encrypt stored data in a user-friendly manner.
FCS_COP.1(d)	Supports	This SFR defines the data encryption algorithm used to protect stored data.
FCS_COP.1(e)	Supports	This SFR defines the key wrap algorithm used to secure the symmetric key that encrypts stored data.
FCS_COP.1(f)	Supports	This SFR defines the key encryption algorithm used to secure the symmetric key that encrypts stored data.
FCS_COP.1(h)	Option	This SFR defines the encryption algorithm used for keyed-hash message authentication.

Objective / SFR	Relationship	Rationale
FCS_COP.1(i)	Supports	This SFR defines the key transport algorithm used for key transport.
FCS_KDF_EXT.1	Option	This SFR defines the key derivation function used by the TOE to ensure that keys are generated in a manner that is not subject to unauthorized disclosure.
FCS_KYC_EXT.1	Satisfies	This SFR defines the key chaining method used by the TOE to provide multiple layers of security for key material.
FCS_PCC_EXT.1	Option	This SFR defines the password-based key derivation function used to construct and condition password data.
FCS_RBG_EXT.1	Supports	This SFR defines the random bit generation algorithm used to ensure that the TOE's cryptographic algorithms function with the theoretical maximum level of security.
FCS_SNI_EXT.1	Selection	This SFR defines secure parameters and methods for salts, nonces, and initialization vectors in order to ensure that cryptographic algorithms operate at their theoretical maximum strength.
FCS_SMC_EXT.1	Selection	This SFR defines appropriate methods of combining submasks that are used to protect the BEV.
FDP_DSK_EXT.1	Satisfies	This SFR requires the TSF to encrypt the data that is stored to disk.
O.AUDIT - ***The TOE shall generate audit data, and be capable of sending it to a trusted External IT Entity. Optionally, it may store audit data in the TOE***.		
FAU_GEN.1	Satisfies	This SFR defines the auditable events for which the TOE generates audit data and the fields that are included in each audit record.
FAU_GEN.2	Satisfies	This SFR defines the ability of the TOE to apply attribution to all activities performed by a user or Administrator.
FAU_SAR.1	Option	This SFR defines the ability of Administrators to read audit data that is stored on the TOE.
FAU_SAR.2	Option	This SFR protects stored audit data from unauthorized access.
FAU_STG.1	Option	This SFR ensures that audit data cannot be modified by untrusted subjects.
FAU_STG.4	Option	This SFR ensures the availability of audit data by taking automatic action in the event the audit storage space is exhausted.
FAU_STG_EXT.1	Satisfies	This SFR defines the ability of the TSF to transmit generated audit data to an external entity using a protected channel
FPT_STM.1	Supports	This SFR ensures that audit data is labeled with accurate timestamps.

Objective / SFR	Relationship	Rationale
FTP_ITC.1	Supports	This SFR defines the protected communications channel(s) over which audit data can be transmitted.
O.TSF_SELF_TEST - ***The TOE shall test some subset of its security functionality to help ensure that subset is operating properly***.		
FPT_TST_EXT.1	Satisfies	This SFR defines the ability of the TSF to perform self-tests which assert the security properties of the TOE.
O.UPDATE_VERIFICATION - ***The TOE shall provide mechanisms to verify the authenticity of software updates***.		
FCS_COP.1(b)	Selection	This SFR defines the digital signature service(s) used to verify the authenticity TOE updates.
FCS_COP.1(c)	Selection	This SFR defines the hashing algorithm(s) used to verify the integrity of TOE updates.
FPT_TUD_EXT.1	Satisfies	This SFR defines the ability of the TOE to be updated and the method(s) by which the updates are known to be trusted.
O.USER_AUTHORIZATION - ***The TOE shall perform authorization of Users in accordance with security policies***.		
FDP_ACC.1	Supports	This SFR enforces User Access Control SFP on subjects, objects, and operations in accordance with user authorization.
FDP_ACF.1	Supports	This SFR enforces the User Access Control SFP to objects based on attributes in accordance with user authorization.
FIA_ATD.1	Supports	This SFR defines the attributes that are associated with Users that can be used to define their authorizations.
FMT_MSA.1	Satisfies	This SFR defines the authorizations that are required to access data that is protected by the TSF.
FMT_MSA.3	Satisfies	This SFR defines the default security posture for enforcement of the access control policy that governs access to data that is protected by the TSF.
FMT_SMF.1	Satisfies	This SFR defines the management functions provided by the TOE that can be used to define User authorizations.
FMT_SMR.1	Satisfies	This SFR defines administrative roles that can be used to define authorizations to groups of Users.
O.USER_I&A - ***The TOE shall perform identification and authentication of Users for operations that require access control, User authorization, or Administrator roles***.		
FIA_AFL.1	Supports	This SFR protects the authentication function by limiting the number of unauthorized authentication attempts that can be made, thereby reducing the likelihood of impersonation.
FIA_PMG_EXT.1	Satisfies	This SFR protects the authentication function by providing for strong credentials that are difficult to guess or derive.

Objective / SFR	Relationship	Rationale
FIA_UAU.1	Satisfies	This SFR defines the TOE functions that can be performed without authentication and the functions that require authentication for use.
FIA_UAU.7	Satisfies	This SFR protects the authentication function by hiding the authentication credential as it is being input.
FIA_UID.1	Satisfies	This SFR defines the TOE functions that can be performed without identification and the functions that require identification for use.
FIA_USB.1	Satisfies	This requirement provides assurance that an identified user is associated with attributes that govern their authorizations to the TSF upon successful authentication to the TOE.
FTA_SSL.3	Satisfies	This SFR helps prevent User or Administrator impersonation by terminating unattended sessions.

Appendix B Conditionally Mandatory Requirements

¶ 923 The following are security functional requirements that are mandatory if the TOE configuration meets the condition(s) specified in section 1.3.1.2.

B.1 Confidential Data on Field-Replaceable Nonvolatile Storage Devices

B.1.1 FPT_KYP_EXT.1 Extended: Protection of Key and Key Material

(for O.KEY_MATERIAL)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 924 **FPT_KYP_EXT.1.1** The TSF shall not store plaintext keys that are part of the keychain specified by FCS_KYC_EXT.1 in **any Field-Replaceable Nonvolatile Storage Device**.

¶ 925 **Assurance Activity:**

¶ 926 ***KMD:***

¶ 927 The evaluator shall examine the Key Management Description (KMD) for a description of the methods used to protect keys stored in nonvolatile memory.

¶ 928 The evaluator shall verify the KMD to ensure it describes the storage location of all keys and the protection of all keys stored in nonvolatile memory.

B.1.2 FCS_KYC_EXT.1 Extended: Key Chaining

(for O.STORAGE_ENCRYPTION)

Hierarchical to: No other components.

Dependencies: [FCS_COP.1(e) Cryptographic operation (Key Wrapping),
FCS_SMC_EXT.1 Extended: Submask Combining,
FCS_COP.1(f) Cryptographic operation (Key Encryption),
FCS_KDF_EXT.1 Cryptographic Operation (Key Derivation),
and/or
FCS_COP.1(i) Cryptographic operation (Key Transport)]

¶ 929 ***Application Note:***

¶ 930 *This SFR forms a keychain that terminates either with a DEK or a BEV to unlock a self-encrypting drive. If passwords are not used, it can be a keychain of*

one, with no intermediate keys forming the DEK or BEV, provided that key is protected. For example, if the DEK for an SED is not stored on the SED and is released on power-up, a keychain of one is allowed.

¶ 931 **FCS_KYC_EXT.1.1** The TSF shall maintain a key chain of: [selection: *one, using a submask as the BEV or DEK; intermediate keys originating from one or more submask(s) to the BEV or DEK using the following method(s):* [selection: *key wrapping as specified in FCS_COP.1(e), key combining as specified in FCS_SMC_EXT.1, key encryption as specified in FCS_COP.1(f), key derivation as specified in FCS_KDF_EXT.1, key transport as specified in FCS_COP.1(i)*]] while maintaining an effective strength of [selection: *128 bits, 256 bits*].

¶ 932 ***Application Note:***

¶ 933 *Key Chaining is the method of using multiple layers of encryption keys to ultimately secure the BEV (Border Encryption Value). The number of intermediate keys will vary – from one (e.g., taking the conditioned password authorization factor and directly using it as the BEV) to many. This applies to all keys that contribute to the ultimate wrapping or derivation of the BEV; including those in areas of protected storage (e.g. TPM stored keys, comparison values).*

¶ 934 *Multiple key chains to the BEV are allowed, as long as all chains meet the key chain requirement.*

¶ 935 *Once the ST Author has selected a method to create the chain (either by unwrapping or encrypting keys), they pull the appropriate requirement out of this appendix. It is allowable for an implementation to use for any or all methods.*

¶ 936 *The method the TOE uses to chain keys and manage/protect them is described in the Key Management Description; see Key Management Description for more information.*

¶ 937 **Assurance activity:**

¶ 938 ***TSS:***

¶ 939 The evaluator shall verify the TSS contains a high-level description of the BEV sizes – that it supports BEV outputs of no fewer 128 bits for products that

support only AES-128, and no fewer than 256 bits for products that support AES-256.

¶ 940 ***KMD:***

¶ 941 The evaluator shall examine the KMD to ensure that it describes a high level description of the key hierarchy for all accepted BEVs. The evaluator shall examine the KMD to ensure it describes the key chain in detail. The description of the key chain shall be reviewed to ensure it maintains a chain of keys using key wrap, submask combining, or key encryption.

¶ 942 The evaluator shall examine the KMD to ensure that it describes how the key chain process functions, such that it does not expose any material that might compromise any key in the chain. (e.g. using a key directly as a compare value against a TPM) This description must include a diagram illustrating the key hierarchy implemented and detail where all keys and keying material is stored or what it is derived from. The evaluator shall examine the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or the initial authorization value and the effective strength of the BEV is maintained throughout the Key Chain.

¶ 943 The evaluator shall verify the KMD includes a description of the strength of keys throughout the key chain.

B.1.3 FDP_DSK_EXT.1 Extended: Protection of Data on Disk

(for O.STORAGE_ENCRYPTION)

Hierarchical to: No other components.

Dependencies: FCS_COP.1(d) Cryptographic operation (AES Data Encryption/Decryption).

¶ 944 **FDP_DSK_EXT.1.1** The TSF shall [selection: *perform encryption in accordance with FCS_COP.1(d) , use a self-encrypting Field-Replaceable Nonvolatile Storage Device that is separately CC certified to conform to the FDE EE cPP*], such that any Field-Replaceable Nonvolatile Storage Device contains no plaintext User Document Data and no plaintext Confidential TSF Data.

¶ 945 ***Application Note:***

¶ 946 *If the self-encrypting device option is selected, the device must be certified in conformance to the current Full Disk Encryption Protection Profile. The ST Author should consult with a CC Scheme for advice on approved Protection Profiles.*

¶ 947 **FDP_DSK_EXT.1.2** The TSF shall encrypt all protected data without user intervention.

¶ 948 ***Application Note:***

¶ 949 *The intent of this requirement is to specify that encryption of any confidential data will not depend on a user electing to protect that data. The 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.*

¶ 950 **Assurance activity:**

¶ 951 In the assurance activities, below, "Device" refers to the Field-Replaceable Nonvolatile Storage Device from FDP_DSK_EXT.1. If the TOE contains more than one applicable Device, then the assurance activities are performed as necessary on each such Device.

¶ 952 ***TSS:***

¶ 953 The evaluator shall examine the TSS to ensure that the description is comprehensive in how the data is written to the Device and the point at which the encryption function is applied.

¶ 954 For the cryptographic functions that are provided by the Operational Environment, the evaluator shall check the TSS to ensure it describes the interface(s) used by the TOE to invoke this functionality.

¶ 955 The evaluator shall verify that the TSS describes the initialization of the Device at shipment of the TOE, or by the activities the TOE performs to ensure that it encrypts all the storage devices entirely when a user or administrator first provisions the Device. The evaluator shall verify the TSS describes areas of the Device that it does not encrypt (e.g., portions that do not contain confidential data boot loaders, partition tables, etc.). If the TOE supports multiple Device encryptions, the evaluator shall examine the administration guidance to ensure the initialization procedure encrypts all Devices.

¶ 956 ***Operational Guidance:***

¶ 957 The evaluator shall review the AGD guidance to determine that it describes the initial steps needed to enable the Device encryption function, including any necessary preparatory steps. The guidance shall provide instructions that are sufficient to ensure that all Devices will be encrypted when encryption is enabled or at shipment of the TOE.

¶ 958 ***KMD:***

¶ 959 The evaluator shall verify the KMD includes a description of the data encryption engine, its components, and details about its implementation (e.g. for hardware: integrated within the device's main SOC or separate co-processor, for software: initialization of the Device, drivers, libraries (if applicable), logical interfaces for encryption/decryption, and areas which are not encrypted (e.g. boot loaders, portions that do not contain confidential data, partition tables, etc.)). The evaluator shall verify the KMD provides a functional (block) diagram showing the main components (such as memories and processors) and the data path between, for hardware, the Device's interface and the Device's persistent media storing the data, or for software, the initial steps needed to the activities the TOE performs to ensure it encrypts the storage device entirely when a user or administrator first provisions the product. The hardware encryption diagram shall show the location of the data encryption engine within the data path. The evaluator shall validate that the hardware encryption diagram contains enough detail showing the main components within the data path and that it clearly identifies the data encryption engine.

¶ 960 The evaluator shall verify the KMD provides sufficient instructions to ensure that when the encryption is enabled, the TOE encrypts all applicable Devices. The evaluator shall verify that the KMD describes the data flow from the interface to the Device's persistent media storing the data. The evaluator shall verify that the KMD provides information on those conditions in which the data bypasses the data encryption engine (e.g. read-write operations to an unencrypted area).

¶ 961 The evaluator shall verify that the KMD provides a description of the boot initialization, the encryption initialization process, and at what moment the product enables the encryption. If encryption can be enabled and disabled, the evaluator shall validate that the product does not allow for the transfer of confidential data before it fully initializes the encryption. The evaluator shall

ensure the software developer provides special tools which allow inspection of the encrypted drive either in-band or out-of-band, and may allow provisioning with a known key.

¶ 962 ***Test:***

¶ 963 The evaluator shall perform the following tests:

¶ 964 Test 1. Write data to Storage device: Perform writing to the storage device with operating TSFI which enforce write process of User documents and Confidential TSF data.

¶ 965 Test 2. Confirm that written data are encrypted: Verify there are no plaintext data present in the encrypted range written by Test 1; and, verify that the data can be decrypted by proper key and key material.

¶ 966 All TSFIs for writing User Document Data and Confidential TSF data should be tested by above Test 1 and Test 2.

B.2 PSTN Fax-Network Separation

B.2.1 FDP_FXS_EXT.1 Extended: Fax separation

(for O.FAX_NET_SEPARATION)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 967 **FDP_FXS_EXT.1.1** The TSF shall prohibit communication via the fax interface, except transmitting or receiving User Data using fax protocols.

¶ 968 ***Application note:***

¶ 969 *FDP_FXS.EXT.1 is required if fax-net separation is performed by the TSF.*

¶ 970 **Assurance Activity:**

¶ 971 The following assurance activities are required when the TOE has a fax communication function to transmit and receive via PSTN.

¶ 972 ***TSS:***

¶ 973 The evaluator shall check the TSS to ensure that it describes:

1. The fax interface use cases
2. The capabilities of the fax modem and the supported fax protocols
3. The data that is allowed to be sent or received via the fax interface
4. How the TOE can only be used transmitting or receiving User Data using fax protocols

¶ 974 ***Operational Guidance:***

¶ 975 The evaluator shall check to ensure that the operational guidance contains a description of the fax interface in terms of usage and available features.

¶ 976 ***Test:***

¶ 977 The evaluator shall test to ensure that the fax interface can only be used transmitting or receiving User Data using fax protocols. Testing will be dependent upon how the TOE enforces this requirement. The following tests shall be used and supplemented with additional testing or a rationale as to why the following tests are sufficient:

1. Verify that the TOE accepts incoming calls using fax carrier protocols and rejects calls that use data carriers. For example, this may be achieved using a terminal application to issue modem commands directly to the TOE from a PC modem (issue terminal command: 'ATDT <TOE Fax Number>') – the TOE should answer the call and disconnect.
2. Verify TOE negotiates outgoing calls using fax carrier protocols and rejects negotiation of data carriers. For example, this may be achieved by using a PC modem to attempt to receive a call from the TOE (submit a fax job from the TOE to <PC modem number>, at PC issue terminal command: 'ATA') – the TOE should disconnect without negotiating a carrier.

Appendix C Optional Requirements

¶ 978 The following are optional security functional requirements and organizational security policies that may be upheld by conforming to the associated security functional requirements.

C.1 Internal Audit Log Storage

¶ 979 The SFRs in this section are to be incorporated in the ST to support the optional Internal Audit Log Storage function.

C.1.1 FAU_SAR.1 Audit review

(for O.AUDIT)

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

¶ 980 **FAU_SAR.1.1** The TSF shall provide [assignment: ***an Administrator***] with the capability to read **all records** from the audit records.

¶ 981 **FAU_SAR.1.2** The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

¶ 982 **Assurance Activity:**

¶ 983 The following assurance activities are required when storing audit records inside the TOE.

¶ 984 ***TSS:***

¶ 985 The evaluator shall check to ensure that the TSS contains a description that audit records can be viewed only by authorized users and functions to view audit records.

¶ 986 The evaluator shall check to ensure that the TSS contains a description of the methods of using interfaces that retrieve audit records (e.g., methods for user identification and authentication, authorization, and retrieving audit records).

¶ 987 ***Operational Guidance:***

¶ 988 The evaluator shall check to ensure that the operational guidance appropriately describes the ways of viewing audit records and forms of viewing.

¶ 989 ***Test:***

¶ 990 The evaluator shall also perform the following tests:

1. The evaluator shall check to ensure that the forms of audit records are provided as specified in the operational guidance by retrieving audit records in accordance with the operational guidance.
2. The evaluator shall check to ensure that no users other than authorized users can retrieve audit records.
3. The evaluator shall check to ensure that all audit records are retrieved by the operation of retrieving audit records.

C.1.2 FAU_SAR.2 Restricted audit review

(for O.AUDIT)

Hierarchical to: No other components.

Dependencies: FAU_SAR.1 Audit review

¶ 991 **FAU_SAR.2.1** The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

¶ 992 **Assurance Activity:**

¶ 993 ***Test:***

¶ 994 The evaluator shall include tests related to this function in the set of tests performed in FMT_SMF.1.

C.1.3 FAU_STG.1 Protected audit trail storage

(for O.AUDIT)

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation

¶ 995 **FAU_STG.1.1** The TSF shall protect the stored audit records in the audit trail from unauthorised deletion.

¶ 996 **FAU_STG.1.2** The TSF shall be able to **prevent** unauthorised modifications to the stored audit records in the audit trail.

¶ 997 **Assurance Activity:**

¶ 998 The following assurance activities are required when storing audit records inside the TOE.

¶ 999 ***TSS:***

¶ 1000 The evaluator shall check to ensure that the TSS contains a description of the means of preventing audit records from unauthorized access (modification, deletion).

¶ 1001 ***Operational Guidance:***

¶ 1002 The evaluator shall check to ensure that the TSS and operational guidance contain descriptions of the interfaces to access to audit records, and if the descriptions of the means of preventing audit records from unauthorized access (modification, deletion) are consistent.

¶ 1003 ***Test:***

¶ 1004 The evaluator shall also perform the following test:

1. The evaluator shall test that an authorized user can access the audit records.
2. The evaluator shall test that a user without authorization for the audit data cannot access the audit records.

C.1.4 FAU_STG.4 Prevention of audit data loss

(for O.AUDIT)

Hierarchical to: FAU_STG.3 Action in case of possible audit data loss

Dependencies: FAU_STG.1 Protected audit trail storage

¶ 1005 **FAU_STG.4.1 Refinement:** The TSF shall [selection, choose one of: ~~***"ignore audited events",***~~ "*prevent audited events, except those taken by the authorised user with special rights*", "*overwrite the oldest stored audit records*"] and [assignment: *other actions to be taken in case of audit storage failure*] if the audit trail is full.

¶ 1006 **Assurance Activity:**

¶ 1007 The following assurance activities are required when storing audit records inside the TOE.

¶ 1008 ***TSS:***

¶ 1009 The evaluator shall check to ensure that the TSS contains a description of the processing performed when the capacity of audit records becomes full, which is consistent with the definition of the SFR.

¶ 1010 ***Operational Guidance:***

¶ 1011 The evaluator shall check to ensure that the operational guidance contains a description of the processing performed (such as informing the authorized users) when the capacity of audit records becomes full.

¶ 1012 ***Test:***

¶ 1013 The evaluator shall also perform the following tests:

1. The evaluator generates auditable events after the capacity of audit records becomes full by generating auditable events in accordance with the operational guidance.
2. The evaluator shall check to ensure that the processing defined in the SFR is appropriately performed to audit records.

C.2 Image Overwrite

¶ 1014 The SFRs in this section are to be incorporated in the ST to support the optional Image Overwrite function.

C.2.1 FDP_RIP.1(a) Subset residual information protection

(for O.IMAGE_OVERWRITE)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 1015 **FDP_RIP.1.1(a) Refinement:** The TSF shall ensure that any previous information content of a resource is made unavailable **by overwriting data** upon the **deallocation of the resource from** the following objects: **D.USER.DOC**.

¶ 1016 **Assurance activity:**

¶ 1017 ***TSS:***

¶ 1018 The evaluator shall examine the TSS to ensure that the description is comprehensive in describing where image data is stored and how and when it is

overwritten.

¶ 1019 ***Operational Guidance:***

¶ 1020 The evaluator shall check to ensure that the operational guidance contains instructions for enabling the Image Overwrite function.

¶ 1021 ***Test:***

¶ 1022 The evaluator shall include tests related to this function in the set of tests performed in FMT_SMF.1.

C.3 Purge Data

¶ 1023 The SFRs in this section are to be incorporated in the ST to support the optional Purge Data function.

C.3.1 FDP_RIP.1(b) Subset residual information protection

(for O.PURGE_DATA)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 1024 **FDP_RIP.1.1(b) Refinement:** The TSF shall ensure that any previous **customer-supplied** information content of a resource is made unavailable upon the **request of an Administrator to** the following objects: **D.USER, D.TSF**.

¶ 1025 **Assurance activity:**

¶ 1026 ***TSS:***

¶ 1027 The evaluator shall examine the TSS to ensure that the description is comprehensive in describing what customer-supplied data is to be purged, where it is stored, and how it is made unavailable.

¶ 1028 ***Operational Guidance:***

¶ 1029 The evaluator shall check to ensure that the operational guidance contains instructions for initiating the Purge Data function.

¶ 1030 ***Test:***

¶ 1031 The evaluator shall include tests related to this function in the set of tests

performed in FMT_SMF.1.

Appendix D Selection-based Requirements

D.1 Confidential Data on Field-Replaceable Nonvolatile Storage Devices

D.1.1 FCS_COP.1(d) Cryptographic operation (AES Data Encryption/Decryption)

(for O. STORAGE_ENCRYPTION)

Hierarchical to: No other components.

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(b) Cryptographic key generation (Symmetric Keys)]

FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

¶ 1032 **FCS_COP.1.1(d)** The TSF shall perform **data encryption and decryption** in accordance with a specified cryptographic algorithm **AES used in [selection: *CBC, GCM, XTS*] mode** and cryptographic key sizes **[selection: *128 bits, 256 bits*]** that meet the following: **AES as specified in ISO/IEC 18033-3, [selection: *CBC as specified in ISO/IEC 10116, GCM as specified in ISO/IEC 19772, and XTS as specified in IEEE 1619*].**

¶ 1033 ***Application Note:***

¶ 1034 *This PP allows for software encryption or hardware encryption.*

¶ 1035 *If XTS Mode is selected, a cryptographic key of 256-bit or of 512-bit is allowed as specified in IEEE 1619. XTS-AES key is divided into two AES keys of equal size - for example, AES-128 is used as the underlying algorithm, when 256-bit key and XTS mode are selected. AES-256 is used when a 512-bit key and XTS mode are selected.*

¶ 1036 *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 Field-Replaceable Nonvolatile Storage Device. 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(b). 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.

¶ 1037 **Assurance activity:**

¶ 1038 ***TSS:***

¶ 1039 The evaluator shall verify the TSS includes a description of the key size used for encryption and the mode used for encryption.

¶ 1040 ***Operational Guidance:***

¶ 1041 If multiple encryption modes are supported, the evaluator examines the guidance documentation to determine that the method of choosing a specific mode/key size by the end user is described.

¶ 1042 ***Test:***

¶ 1043 The following tests are conditional based upon the selections made in the SFR.

¶ 1044 **AES-CBC Tests**

¶ 1045 AES-CBC Known Answer Tests

¶ 1046 There are four Known Answer Tests (KATs), described below. In all KATs, the plaintext, ciphertext, and IV values shall be 128-bit blocks. The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

¶ 1047 **KAT-1.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 plaintext values and obtain the ciphertext value that results from AES-CBC encryption of the given plaintext using a key value of all zeros and an IV of all zeros. Five plaintext values shall be encrypted with a 128-bit all-zeros key, and the other five shall be encrypted with a 256-bit all-zeros key.

¶ 1048 To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using 10 ciphertext values as input and AES-CBC decryption.

¶ 1049 **KAT-2.** To test the encrypt functionality of AES-CBC, the evaluator shall supply a set of 10 key values and obtain the ciphertext value that results from

AES-CBC encryption of an all-zeros plaintext using the given key value and an IV of all zeros. Five of the keys shall be 128-bit keys, and the other five shall be 256-bit keys.

¶ 1050 To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using an all-zero ciphertext value as input and AES-CBC decryption.

¶ 1051 **KAT-3.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the two sets of key values described below and obtain the ciphertext value that results from AES encryption of an all-zeros plaintext using the given key value and an IV of all zeros. The first set of keys shall have 128 128-bit keys, and the second set shall have 256 256-bit keys. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N].

¶ 1052 To test the decrypt functionality of AES-CBC, the evaluator shall supply the two sets of key and ciphertext value pairs described below and obtain the plaintext value that results from AES-CBC decryption of the given ciphertext using the given key and an IV of all zeros. The first set of key/ciphertext pairs shall have 128 128-bit key/ciphertext pairs, and the second set of key/ciphertext pairs shall have 256 256-bit key/ciphertext pairs. Key i in each set shall have the leftmost i bits be ones and the rightmost N-i bits be zeros, for i in [1,N]. The ciphertext value in each pair shall be the value that results in an all-zeros plaintext when decrypted with its corresponding key.

¶ 1053 **KAT-4.** To test the encrypt functionality of AES-CBC, the evaluator shall supply the set of 128 plaintext values described below and obtain the two ciphertext values that result from AES-CBC encryption of the given plaintext using a 128-bit key value of all zeros with an IV of all zeros and using a 256-bit key value of all zeros with an IV of all zeros, respectively. Plaintext value i in each set shall have the leftmost i bits be ones and the rightmost 128-i bits be zeros, for i in [1,128].

¶ 1054 To test the decrypt functionality of AES-CBC, the evaluator shall perform the same test as for encrypt, using ciphertext values of the same form as the plaintext in the encrypt test as input and AES-CBC decryption.

¶ 1055 AES-CBC Multi-Block Message Test

¶ 1056 The evaluator shall test the encrypt functionality by encrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and plaintext message of length i blocks and encrypt the message, using the mode to be tested, with the chosen key and IV. The ciphertext shall be compared to the result of encrypting the same plaintext message with the same key and IV using a known good implementation.

¶ 1057 The evaluator shall also test the decrypt functionality for each mode by decrypting an i-block message where 1 < i <=10. The evaluator shall choose a key, an IV and a ciphertext message of length i blocks and decrypt the message, using the mode to be tested, with the chosen key and IV. The plaintext shall be compared to the result of decrypting the same ciphertext message with the same key and IV using a known good implementation.

¶ 1058 AES-CBC Monte Carlo Tests

¶ 1059 The evaluator shall test the encrypt functionality using a set of 200 plaintext, IV, and key 3-tuples. 100 of these shall use 128 bit keys, and 100 shall use 256 bit keys. The plaintext and IV values shall be 128-bit blocks. For each 3-tuple, 1000 iterations shall be run as follows:

```
¶ 1060 # Input: PT, IV, Key
¶ 1061 for i = 1 to 1000:
¶ 1062         if i == 1:
¶ 1063             CT[1] = AES-CBC-Encrypt(Key, IV, PT)
¶ 1064             PT = IV
¶ 1065         else:
¶ 1066             CT[i] = AES-CBC-Encrypt(Key, PT)
¶ 1067             PT = CT[i-1]
```

¶ 1068 The ciphertext computed in the 1000th iteration (i.e., CT[1000]) is the result for that trial. This result shall be compared to the result of running 1000 iterations with the same values using a known good implementation.

¶ 1069 The evaluator shall test the decrypt functionality using the same test as for encrypt, exchanging CT and PT and replacing AES-CBC-Encrypt with AES-CBC-Decrypt.

¶ 1070 AES-GCM Test

¶ 1071 The evaluator shall test the authenticated encrypt functionality of AES-GCM for each combination of the following input parameter lengths:

¶ 1072 128 bit and 256 bit keys

¶ 1073 **Two plaintext lengths**. One of the plaintext lengths shall be a non-zero integer multiple of 128 bits, if supported. The other plaintext length shall not be an integer multiple of 128 bits, if supported.

¶ 1074 **Three AAD lengths**. One AAD length shall be 0, if supported. One AAD length shall be a non-zero integer multiple of 128 bits, if supported. One AAD length shall not be an integer multiple of 128 bits, if supported.

¶ 1075 **Two IV lengths**. If 96 bit IV is supported, 96 bits shall be one of the two IV lengths tested.

¶ 1076 The evaluator shall test the encrypt functionality using a set of 10 key, plaintext, AAD, and IV tuples for each combination of parameter lengths above and obtain the ciphertext value and tag that results from AES-GCM authenticated encrypt. Each supported tag length shall be tested at least once per set of 10. The IV value may be supplied by the evaluator or the implementation being tested, as long as it is known.

¶ 1077 The evaluator shall test the decrypt functionality using a set of 10 key, ciphertext, tag, AAD, and IV 5-tuples for each combination of parameter lengths above and obtain a Pass/Fail result on authentication and the decrypted plaintext if Pass. The set shall include five tuples that Pass and five that Fail.

¶ 1078 The results from each test may either be obtained by the evaluator directly or by supplying the inputs to the implementer and receiving the results in response. To determine correctness, the evaluator shall compare the resulting values to those obtained by submitting the same inputs to a known good implementation.

¶ 1079 XTS-AES Test

¶ 1080 The evaluator shall test the encrypt functionality of XTS-AES for each combination of the following input parameter lengths:

¶ 1081 256 bit (for AES-128) and 512 bit (for AES-256) keys

¶ 1082 **Three data unit (i.e., plaintext) lengths**. One of the data unit lengths shall be a non-zero integer multiple of 128 bits, if supported. One of the data unit lengths

shall be an integer multiple of 128 bits, if supported. The third data unit length shall be either the longest supported data unit length or 2^{16} bits, whichever is smaller.

¶ 1083 The evaluator shall test the encrypt functionality using a set of 100 (key, plaintext and 128-bit random tweak value) 3-tuples and obtain the ciphertext that results from XTS-AES encrypt.

¶ 1084 The evaluator may supply a data unit sequence number instead of the tweak value if the implementation supports it. The data unit sequence number is a base-10 number ranging between 0 and 255 that implementations convert to a tweak value internally.

¶ 1085 The evaluator shall test the decrypt functionality of XTS-AES using the same test as for encrypt, replacing plaintext values with ciphertext values and XTS-AES encrypt with XTS-AES decrypt.

D.1.2 FCS_COP.1(e) Cryptographic operation (Key Wrapping)

(selected in FCS_KYC_EXT.1.1)

Hierarchical to: No other components.

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(b) Cryptographic key generation (Symmetric Keys)]

FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

¶ 1086 **FCS_COP.1.1(e) Refinement:** The TSF shall perform **key wrapping** in accordance with a specified cryptographic algorithm **AES in the following modes [selection: *KW, KWP, GCM, CCM*]** and the cryptographic key size **[selection: *128 bits, 256 bits***] that meet the following: [**ISO/IEC 18033-3 (AES), [selection: *NIST SP 800-38F, ISO/IEC 19772***]].

¶ 1087 ***Application Note:***

¶ 1088 *This requirement is used in the body of the ST if the ST Author chooses to use key wrapping in the key chaining approach that is specified in FCS_KYC_EXT.1.*

¶ 1089 **Assurance activity:**

¶ 1090 ***TSS:***

¶ 1091 The evaluator shall verify the TSS includes a description of the key wrap function(s) and shall verify the key wrap uses an approved key wrap algorithm according to the appropriate specification.

¶ 1092 ***KMD:***

¶ 1093 The evaluator shall review the KMD to ensure that all keys are wrapped using the approved method and a description of when the key wrapping occurs.

¶ 1094 ***Test:***

¶ 1095 The evaluator shall ensure the wrapped key is wrapped as specified in this SFR using reference implementation of wrapping in accordance with AES in the modes and key size specified in this SFR. This reference implementation of wrapping algorithm may be a tool or program provided by the evaluator or the developer, this implementation is dependent on the KMD description provided by the developer.

D.1.3 FCS_COP.1(f) Cryptographic operation (Key Encryption)

(selected from FCS_KYC_EXT.1.1)

Hierarchical to: No other components.

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(b) Cryptographic key generation (Symmetric Keys)]

FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

¶ 1096 **FCS_COP.1.1(f) Refinement:** The TSF shall perform **key encryption and decryption** in accordance with a specified cryptographic algorithm **AES used in [[selection: *CBC, GCM*] mode]** and cryptographic key sizes **[selection: *128 bits, 256 bits*]** that meet the following: [**AES as specified in ISO /IEC 18033-3, [selection: *CBC as specified in ISO/IEC 10116, GCM as specified in ISO/IEC 19772***].

¶ 1097 ***Application Note:***

¶ 1098 *This requirement is used in the body of the ST if the ST Author chooses to use AES encryption/decryption for protecting the keys as part of the key chaining approach that is specified in FCS_KYC_EXT.1.*

¶ 1099 **Assurance activity:**

¶ 1100 ***TSS:***

¶ 1101 The evaluator shall verify the TSS includes a description of the key encryption function(s) and shall verify the key encryption uses an approved algorithm according to the appropriate specification.

¶ 1102 ***KMD:***

¶ 1103 The evaluator shall review the KMD to ensure that all keys are encrypted using the approved method and a description of when the key encryption occurs is provided.

¶ 1104 ***Test:***

¶ 1105 The evaluator shall use tests in FCS_COP.1(d) to verify encryption.

D.1.4 FCS_COP.1(i) Cryptographic operation (Key Transport)

(selected in FCS_KYC_EXT.1.1)

Hierarchical to: No other components.

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(a) Cryptographic Key Generation (for asymmetric keys)]

FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

¶ 1106 **FCS_COP.1.1(i) Refinement:** The TSF shall perform **key transport** in accordance with a specified cryptographic algorithm **RSA in the following modes [selection: *KTS-OAEP, KTS-KEM-KWS*]** and the cryptographic key size **[selection: *2048, 3072*]** that meet the following**: NIST SP 800-56B, Revision 1**.

¶ 1107 ***Application Note:***

¶ 1108 *This requirement is used in the body of the ST if the ST Author chooses to use key transport in the key chaining approach that is specified in FCS_KYC_EXT.1.*

D.1.5 FCS_SMC_EXT.1 Extended: Submask Combining

(selected in FCS_KYC_EXT.1.1)

Hierarchical to: No other components.

Dependencies: FCS_COP.1(c) Cryptographic operation (Hash Algorithm)

¶ 1109 **FCS_SMC_EXT.1.1** The TSF shall combine submasks using the following method [selection: *exclusive OR (XOR), SHA-256, SHA-512*] to generate an intermediary key or BEV.

¶ 1110 ***Application Note:***

¶ 1111 *This requirement specifies the way that a product may combine the various submasks by using either an XOR or an approved SHA-hash. The approved hash function is captured in FCS_COP.1(c) in Appendix D.3.1.*

¶ 1112 **Assurance activity:**

¶ 1113 ***TSS:***

¶ 1114 If keys are XORed together to form an intermediate key, 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.

¶ 1115 ***KMD:***

¶ 1116 The evaluator shall review the KMD to ensure that an approved combination is used and does not result in the weakening or exposure of key material.

¶ 1117 ***Test:***

¶ 1118 (conditional): If there is more than one authorization factor, the evaluator shall ensure that failure to supply a required authorization factor does not result in access to the encrypted data.

D.2 Protected Communications

¶ 1119 As indicated in the FTP requirements, there are several methods by which conformant

TOEs can mitigate threats against compromise of the communication channel between administrators, other portions of the TOE, or external IT entities. One of the secure communication protocols (IPsec, SSH, TLS, TLS/HTTPS) must be implemented in order to provide protected connectivity for (at a minimum) the audit server and remote administrators.

¶ 1120 There are unique requirements associated with each of the protocol suites; these are specified in below. Depending on the selections for the FTP_ITC.1 and FTP_TRP.1 components, the ST author will need to include the associated SFRs and Assurance Activities in the ST.

D.2.1 FCS_IPSEC_EXT.1 Extended: IPsec selected

(selected in FTP_ITC.1.1, FTP_TRP.1.1)

Hierarchical to: No other components.

Dependencies: ~~FPT_ITT.1 Basic internal TSF data transfer protection~~,

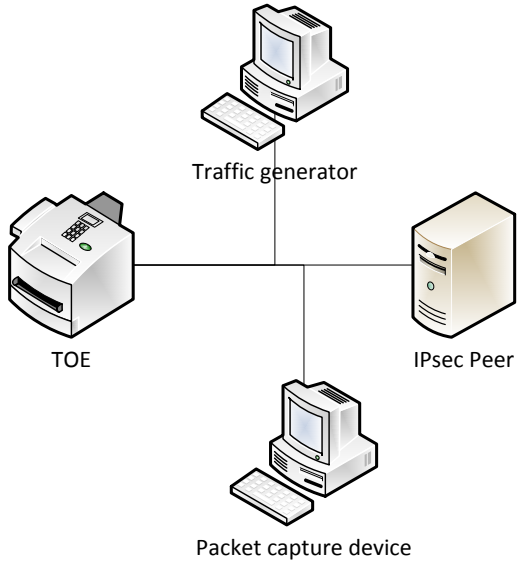
FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition

FCS_COP.1(g) Cryptographic Operation (for keyed-hash message authentication)

¶ 1121 ***Application Note:***

¶ 1122 *In order to show that the TSF implements the RFCs in accordance with the requirements of this PP, the evaluator shall perform the assurance activities listed below.*

¶ 1123 *The TOE is required to use the IPsec protocol to establish connections used to communicate with an IPsec Peer.*



¶ 1124

¶ 1125 The evaluators shall minimally create a test environment equivalent to the test environment illustrated above. It is expected that the traffic generator is used to construct network packets and will provide the evaluator with the ability manipulate fields in the ICMP, IPv4, IPv6, UDP, and TCP packet headers. The evaluators must provide justification for any differences in the test environment.

¶ 1126 **FCS_IPSEC_EXT.1.1** The TSF shall implement the IPsec architecture as specified in RFC 4301.

¶ 1127 **Assurance Activity:**

¶ 1128 ***Operational Guidance:***

¶ 1129 The evaluator shall examine the operational guidance to verify it instructs the Administrator how to construct entries into the SPD that specify a rule for DISCARD, BYPASS and PROTECT.

¶ 1130 ***Test:***

¶ 1131 The evaluator uses the operational guidance to configure the TOE to carry out the following tests:

1. The evaluator shall configure the SPD such that there is a rule for DISCARD, BYPASS, PROTECT. The selectors used in the construction of the rule shall be different such that the evaluator can send in three network packets with the appropriate fields in the packet header that each packet will match one of the three rules. The evaluator observes via the

audit trail, and packet captures that the TOE exhibited the expected behavior: appropriate packet was dropped, allowed through without modification, was encrypted by the IPsec implementation.

2. The evaluator shall devise two equal SPD entries with alternate operations – BYPASS and PROTECT. The entries should then be deployed in two distinct orders and in each case the evaluator shall ensure that the first entry is enforced in both cases by generating applicable packets and using packet capture and logs for confirmation.

3. The evaluator shall repeat the procedure above, except that the two entries should be devised where one is a subset of the other (e.g., a specific address vs. a network segment). Again, the evaluator should test both orders to ensure that the first is enforced regardless of the specificity of the rule.

¶ 1132 **FCS_IPSEC_EXT.1.2** The TSF shall implement [selection: *tunnel mode*, *transport mode*].

¶ 1133 **Assurance Activity:**

¶ 1134 ***TSS:***

¶ 1135 The evaluator checks the TSS to ensure it states that the VPN can be established to operate in tunnel mode and/or transport mode (as selected).

¶ 1136 ***Operational Guidance:***

¶ 1137 The evaluator shall confirm that the operational guidance contains instructions on how to configure the connection in each mode selected.

¶ 1138 ***Test:***

¶ 1139 The evaluator shall perform the following test(s) based on the selections chosen:

1. (conditional): If tunnel mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in tunnel mode and also configures an IPsec Peer to operate in tunnel mode. The evaluator configures the TOE and the IPsec Peer to use any of the allowable cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator shall then initiate a

connection from the client to connect to the IPsec Peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the tunnel mode.

2. (conditional): If transport mode is selected, the evaluator uses the operational guidance to configure the TOE to operate in transport mode and also configures an IPsec Peer to operate in transport mode. The evaluator configures the TOE and the IPsec Peer to use any of the allowed cryptographic algorithms, authentication methods, etc. to ensure an allowable SA can be negotiated. The evaluator then initiates a connection from the TOE to connect to the IPsec Peer. The evaluator observes (for example, in the audit trail and the captured packets) that a successful connection was established using the transport mode.

¶ 1140 **FCS_IPSEC_EXT.1.3** The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

¶ 1141 **Assurance Activity:**

¶ 1142 ***TSS:***

¶ 1143 The evaluator shall examine the TSS to verify that the TSS provides a description of how a packet is processed against the SPD and that if no “rules” are found to match, that a final rule exists, either implicitly or explicitly, that causes the network packet to be discarded.

¶ 1144 ***Operational Guidance:***

¶ 1145 The evaluator checks that the operational guidance provides instructions on how to construct the SPD and uses the guidance to configure the TOE for the following tests.

¶ 1146 ***Test:***

¶ 1147 The evaluator shall perform the following test:

¶ 1148 The evaluator shall configure the SPD such that it has entries that contain operations that DISCARD, BYPASS, and PROTECT network packets. The evaluator may use the SPD that was created for verification of FCS_IPSEC_EXT.1.1. The evaluator shall construct a network packet that matches a BYPASS entry and send that packet. The evaluator should observe

that the network packet is passed to the proper destination interface with no modification. The evaluator shall then modify a field in the packet header; such that it no longer matches the evaluator-created entries (there may be a "TOE created" final entry that discards packets that do not match any previous entries). The evaluator sends the packet, and observes that the packet was not permitted to flow to any of the TOE's interfaces.

¶ 1149 **FCS_IPSEC_EXT.1.4** The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using [selection: *the cryptographic algorithms AES-CBC-128 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, AES-CBC-256 (as specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, AES-GCM-128 as specified in RFC 4106, AES-GCM-256 as specified in RFC 4106*].

¶ 1150 **Assurance Activity:**

¶ 1151 ***TSS:***

¶ 1152 The evaluator shall examine the TSS to verify that the symmetric encryption algorithms selected (along with the SHA-based HMAC algorithm, if AES-CBC is selected) are described. If selected, the evaluator ensures that the SHA-based HMAC algorithm conforms to the algorithms specified in FCS_COP.1(g) Cryptographic Operations (for keyed-hash message authentication).

¶ 1153 ***Operational Guidance:***

¶ 1154 The evaluator checks the operational guidance to ensure it provides instructions on how to configure the TOE to use the algorithms selected by the ST author.

¶ 1155 ***Test:***

¶ 1156 The evaluator shall also perform the following tests:

¶ 1157 The evaluator shall configure the TOE as indicated in the operational guidance configuring the TOE to using each of the selected algorithms, and attempt to establish a connection using ESP. The connection should be successfully established for each algorithm.

¶ 1158 **FCS_IPSEC_EXT.1.5** The TSF shall implement the protocol: [selection: *IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109,* [selection: *no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers*], and [selection: *no other RFCs for hash functions, RFC 4868 for hash*

functions]; *IKEv2 as defined in RFCs 5996,* [selection: *with no support for NAT traversal, with mandatory support for NAT traversal as specified in section 2.23*], and [selection: *no other RFCs for hash functions, RFC 4868 for hash functions*]].

¶ 1159 ***Application Note:***

¶ 1160 *Either IKEv1 or IKEv2 support must be provided, although conformant TOEs can provide both; the first selection is used to make this choice. For IKEv1, the requirement is to be interpreted as requiring the IKE implementation conforming to RFC 2409 with the additions/modifications as described in RFC 4109. RFC 4304 identifies support for extended sequence numbers, which compliant TOEs can specify using the second selection. RFC 4868 identifies additional hash functions for use with both IKEv1 and IKEv2; if these functions are implemented, the third (for IKEv1) and fourth (for IKEv2) selection can be used.*

¶ 1161 **Assurance Activity:**

¶ 1162 ***TSS:***

¶ 1163 The evaluator shall examine the TSS to verify that IKEv1 and/or IKEv2 are implemented.

¶ 1164 ***Operational Guidance:***

¶ 1165 The evaluator shall check the operational guidance to ensure it instructs the administrator how to configure the TOE to use IKEv1 and/or IKEv2 (as selected), and uses the guidance to configure the TOE to perform NAT traversal for the following test if IKEv2 is selected.

¶ 1166 ***Test:***

¶ 1167 (conditional): If IKEv2 is selected, the evaluator shall configure the TOE so that it will perform NAT traversal processing as described in the TSS and RFC 5996, section 2.23. The evaluator shall initiate an IPsec connection and determine that the NAT is successfully traversed.

¶ 1168 **FCS_IPSEC_EXT.1.6** The TSF shall ensure the encrypted payload in the [selection: *IKEv1, IKEv2*] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 3602 and [selection: *AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm*].

¶ 1169 **Assurance Activity:**

¶ 1170 ***TSS:***

¶ 1171 The evaluator shall ensure the TSS identifies the algorithms used for encrypting the IKEv1 and/or IKEv2 payload, and that the algorithms AES-CBC-128, AES-CBC-256 are specified, and if others are chosen in the selection of the requirement, those are included in the TSS discussion.

¶ 1172 ***Operational Guidance:***

¶ 1173 The evaluator ensures that the operational guidance describes the configuration of the mandated algorithms, as well as any additional algorithms selected in the requirement. The guidance is then used to configure the TOE to perform the following test for each ciphersuite selected.

¶ 1174 ***Test:***

¶ 1175 The evaluator shall configure the TOE to use the ciphersuite under test to encrypt the IKEv1 and/or IKEv2 payload and establish a connection with a peer device, which is configured to only accept the payload encrypted using the indicated ciphersuite. The evaluator will confirm the algorithm was that used in the negotiation.

¶ 1176 **FCS_IPSEC_EXT.1.7** The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

¶ 1177 **Assurance Activity:**

¶ 1178 ***TSS:***

¶ 1179 The evaluator shall examine the TSS to ensure that, in the description of the IPsec protocol supported by the TOE, it states that aggressive mode is not used for IKEv1 Phase 1 exchanges, and that only main mode is used. It may be that this is a configurable option.

¶ 1180 ***Operational Guidance:***

¶ 1181 If the mode requires configuration of the TOE prior to its operation, the evaluator shall check the operational guidance to ensure that instructions for this configuration are contained within that guidance.

¶ 1182 ***Test:***

¶ 1183 The evaluator shall also perform the following test:

¶ 1184 (conditional): The evaluator shall configure the TOE as indicated in the operational guidance, and attempt to establish a connection using an IKEv1 Phase 1 connection in aggressive mode. This attempt should fail. The evaluator should then show that main mode exchanges are supported. This test is not applicable if IKEv1 is not selected above in the FCS_IPSEC_EXT.1.5 protocol selection.

¶ 1185 **FCS_IPSEC_EXT.1.8** The TSF shall ensure that [selection: *IKEv2 SA lifetimes can be established based on* [selection: *number of packets/number of bytes; length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs*]; *IKEv1 SA lifetimes can be established based on [selection: number of packets/number of bytes ; length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs*]].

¶ 1186 ***Application Note:***

¶ 1187 *The ST Author is afforded a selection based on the version of IKE in their implementation. If the lifetime limitations are configurable, then the evaluator verifies that the appropriate instructions for configuring these values are included in the operational guidance.*

¶ 1188 *As far as SA lifetimes are concerned, the TOE can limit the lifetime based on the number of bytes transmitted, or the number of packets transmitted. Either packet-based or volume-based SA lifetimes are acceptable; the ST author makes the appropriate selection to indicate which type of lifetime limits are supported.*

¶ 1189 **Assurance Activity:**

¶ 1190 ***Operational Guidance:***

¶ 1191 The evaluator verifies that the values for SA lifetimes can be configured and that the instructions for doing so are located in the operational guidance. If time-based limits are supported, the evaluator ensures that the values allow for Phase 1 SAs values for 24 hours and 8 hours for Phase 2 SAs. Currently there are no values mandated for the number of packets or number of bytes, the evaluator just ensures that this can be configured if selected in the requirement.

¶ 1192 When testing this functionality, the evaluator needs to ensure that both sides are

configured appropriately. From the RFC "A difference between IKEv1 and IKEv2 is that in IKEv1 SA lifetimes were negotiated. In IKEv2, each end of the SA is responsible for enforcing its own lifetime policy on the SA and rekeying the SA when necessary. If the two ends have different lifetime policies, the end with the shorter lifetime will end up always being the one to request the rekeying. If the two ends have the same lifetime policies, it is possible that both will initiate a rekeying at the same time (which will result in redundant SAs). To reduce the probability of this happening, the timing of rekeying requests SHOULD be jittered."

¶ 1193 ***Test:***

¶ 1194 Each of the following tests shall be performed for each version of IKE selected in the FCS_IPSEC_EXT.1.5 protocol selection:

1. (Conditional): The evaluator shall configure a maximum lifetime in terms of the # of packets (or bytes) allowed following the operational guidance. The evaluator shall establish an SA and determine that once the allowed # of packets (or bytes) through this SA is exceeded, the connection is renegotiated.

2. (Conditional): The evaluator shall construct a test where a Phase 1 SA is established and attempted to be maintained for more than 24 hours before it is renegotiated. The evaluator shall observe that this SA is closed or renegotiated in 24 hours or less. If such an action requires that the TOE be configured in a specific way, the evaluator shall implement tests demonstrating that the configuration capability of the TOE works as documented in the operational guidance.

3. (Conditional): The evaluator shall perform a test similar to Test 1 for Phase 2 SAs, except that the lifetime will be 8 hours instead of 24.

¶ 1195 **FCS_IPSEC_EXT.1.9** The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [selection: *24 (2048-bit MODP with 256-bit POS), 19 (256-bit Random ECP), 20 (384-bit Random ECP, 5 (1536-bit MODP)),* [assignment: *other DH groups that are implemented by the TOE], no other DH groups*].

¶ 1196 ***Application Note:***

¶ 1197 *The above requires that the TOE support DH Group 14. If other groups are supported, then those should be selected (for groups 24, 19, 20, and 5) or specified in the assignment above; otherwise "no other DH groups" should be selected. This applies to IKEv1/IKEv2 exchanges.*

¶ 1198 **Assurance Activity:**

¶ 1199 ***TSS:***

¶ 1200 The evaluator shall check to ensure that the DH groups specified in the requirement are listed as being supported in the TSS. If there is more than one DH group supported, the evaluator checks to ensure the TSS describes how a particular DH group is specified/negotiated with a peer.

¶ 1201 ***Test:***

¶ 1202 The evaluator shall also perform the following test (this test may be combined with other tests for this component, for instance, the tests associated with FCS_IPSEC_EXT.1.1):

¶ 1203 For each supported DH group, the evaluator shall test to ensure that all IKE protocols can be successfully completed using that particular DH group.

¶ 1204 **FCS_IPSEC_EXT.1.10** The TSF shall ensure that all IKE protocols perform Peer Authentication using the [selection: *RSA, ECDSA*] algorithm and Pre-shared Keys.

¶ 1205 ***Application Note:***

¶ 1206 *The selected algorithm should correspond to an appropriate selection for FCS_COP.1(b). If IPsec is included in the TOE, the ST author also includes FIA_PSK_EXT from Appendix D.2.6.*

¶ 1207 **Assurance Activity:**

¶ 1208 ***TSS:***

¶ 1209 The evaluator shall check that the TSS contains a description of the IKE peer authentication process used by the TOE, and that this description covers the use of the signature algorithm or algorithms specified in the requirement.

¶ 1210 ***Test:***

¶ 1211 The evaluator shall also perform the following test:

¶ 1212 For each supported signature algorithm, the evaluator shall test that peer authentication using that algorithm can be successfully achieved and results in the successful establishment of a connection.

D.2.2 FCS_TLS_EXT.1 Extended: TLS selected

(selected in FTP_ITC.1.1, FTP_TRP.1.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 1213 **FCS_TLS_EXT.1.1** The TSF shall implement one or more of the following protocols [selection: *TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)*] supporting the following ciphersuites:

¶ 1214 Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA

¶ 1215 Optional Ciphersuites:

¶ 1216 [selection:

- *None*
- *TLS_RSA_WITH_AES_256_CBC_SHA*
- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA*
- *TLS_RSA_WITH_AES_128_CBC_SHA256*
- *TLS_RSA_WITH_AES_256_CBC_ SHA256*
- *TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256*
- *TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA*

- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384*
- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384*

¶ 1217].

¶ 1218 ***Application Note:***

¶ 1219 *The ST author must make the appropriate selections and assignments to reflect the TLS implementation.*

¶ 1220 *The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the optional ciphersuites that are supported; if there are no ciphersuites supported other than the mandatory suites, then "None" should be selected. If administrative steps need to be taken so that the suites negotiated by the implementation are limited to those in this requirement, the appropriate instructions need to be contained in the guidance called for by AGD_OPE.*

¶ 1221 *The Suite B algorithms (RFC 5430) listed above are the preferred algorithms for implementation. The TLS requirement may be changed in the next version of the HCD PP to comply with CNSSP 15 and NIST SP 800-131A.*

¶ 1222 **Assurance Activity:**

¶ 1223 ***TSS:***

¶ 1224 The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that

TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).

¶ 1225 ***Test:***

¶ 1226 The evaluator shall also perform the following test:

1. The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of a HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES).
2. The evaluator shall setup a man-in-the-middle tool between the TOE and the TLS Peer and shall perform the following modifications to the traffic:
    a. [Conditional: TOE is a server] Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the server denies the client's Finished handshake message.
    b. [Conditional: TOE is a client] Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello.
    c. [Conditional: TOE is a client] If a DHE or ECDHE ciphersuite is supported, modify the signature block in the Server's KeyExchange handshake message, and verify that the client rejects the connection after receiving the Server KeyExchange.
    d. [Conditional: TOE is a client] Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.

D.2.3 FCS_SSH_EXT.1 Extended: SSH selected

(selected in FTP_ITC.1.1, FTP_TRP.1.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 1227 **FCS_SSH_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: *5656, 6668, no other RFCs*].

¶ 1228 ***Application Note:***

¶ 1229 *The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted).*

¶ 1230 *In the next version of this PP, a requirement may be added regarding rekeying. The requirement would read "The TSF shall ensure that the SSH connection be rekeyed after no more than 228 packets have been transmitted using that key."*

¶ 1231 **FCS_SSH_EXT.1.2** The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, password-based.

¶ 1232 **Assurance Activity:**

¶ 1233 ***TSS:***

¶ 1234 The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.5, and ensure that password-based authentication methods are also allowed.

¶ 1235 ***Test:***

¶ 1236 The evaluator shall also perform the following tests:

1. The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance.
2. Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.

¶ 1237 **FCS_SSH_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [assignment: *number of bytes*] bytes in an SSH transport connection are dropped.

¶ 1238 ***Application Note:***

¶ 1239 *RFC 4253 provides for the acceptance of "large packets" with the caveat that the packets should be of "reasonable length" or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining "reasonable length" for the TOE.*

¶ 1240 **Assurance Activity:**

¶ 1241 ***Test:***

¶ 1242 The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.

¶ 1243 **FCS_SSH_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [selection: *AEAD_AES_128_GCM, AEAD_AES_256_GCM, no other algorithms*].

¶ 1244 ***Application Note:***

¶ 1245 *In the assignment, the ST author can select the AES-GCM algorithms, or "no other algorithms" if AES-GCM is not supported. If AES-GCM is selected, there should be corresponding FCS_COP entries in the ST.*

¶ 1246 **Assurance Activity:**

¶ 1247 ***TSS:***

¶ 1248 The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements).

¶ 1249 ***Test:***

¶ 1250 The evaluator shall also perform the following test:

¶ 1251 The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

¶ 1252 **FCS_SSH_EXT.1.5** The TSF shall ensure that the SSH transport implementation uses [selection: SSH_RSA, ecdsa-sha2-nistp256] and [selection: PGP-SIGN-RSA, PGP-SIGN-DSS, ecdsa-sha2-nistp384, no other public key algorithms,] as its public key algorithm(s).

¶ 1253 **Assurance Activity:**

¶ 1254 The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.

¶ 1255 **FCS_SSH_EXT.1.6** The TSF shall ensure that data integrity algorithms used in SSH transport connection is [selection: *HMAC-SHA1, HMAC-SHA1-96, HMAC-SHA2-256, HMAC-SHA2-512*].

¶ 1256 ***Application Note:***

¶ 1257 *RFC 6668 specifies the use of the SHA-2 algorithms in SSH.*

¶ 1258 **Assurance Activity:**

¶ 1259 ***TSS:***

¶ 1260 The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component. The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the “none” MAC algorithm is not allowed).

¶ 1261 ***Test:***

¶ 1262 The evaluator shall also perform the following test:

¶ 1263 The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.

¶ 1264 **FCS_SSH_EXT.1.7** The TSF shall ensure that diffie-hellman-group14-sha1 and [selection: ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange method used for the SSH protocol.

¶ 1265 **Assurance Activity:**

¶ 1266 ***Operational Guidance:***

¶ 1267 The evaluator shall ensure that operational guidance contains configuration information that will allow the security administrator to configure the TOE so that all key exchanges for SSH are performed using DH group 14 and any groups specified from the selection in the ST. If this capability is "hard-coded" into the TOE, the evaluator shall check the TSS to ensure that this is stated in the discussion of the SSH protocol.

¶ 1268 ***Test:***

¶ 1269 The evaluator shall also perform the following test:

¶ 1270 The evaluator shall attempt to perform a diffie-hellman-group1-sha1 key exchange, and observe that the attempt fails. For each allowed key exchange method, the evaluator shall then attempt to perform a key exchange using that method, and observe that the attempt succeeds.

D.2.4 FCS_HTTPS_EXT.1 Extended: HTTPS selected

(selected in FTP_ITC.1.1, FTP_TRP.1.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 1271 **FCS_HTTPS_EXT.1.1** The TSF shall implement the HTTPS protocol that complies with RFC 2818.

¶ 1272 ***Application Note:***

¶ 1273 *The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done either by adding elements to this component, or by additional detail in the TSS.*

¶ 1274 **FCS_HTTPS_EXT.1.2** The TSF shall implement HTTPS using TLS as specified in

FCS_TLS_EXT.1.

¶ 1275 **Assurance Activity:**

¶ 1276 ***TSS:***

¶ 1277 The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish an administrative session, focusing on any client authentication required by the TLS protocol vs. security administrator authentication which may be done at a different level of the processing stack.

¶ 1278 ***Test:***

¶ 1279 Testing for this activity is done as part of the TLS testing; this may result in additional testing if the TLS tests are done at the TLS protocol level.

D.2.5 FCS_COP.1(g) Cryptographic Operation (for keyed-hash message authentication)

(selected with FCS_IPSEC_EXT.1.4)

Hierarchical to: No other components.

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(b) Cryptographic key generation (Symmetric Keys)]

FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

¶ 1280 **FCS_COP.1.1(g) Refinement**: The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-[selection: *SHA-1, SHA-224, SHA-256, SHA-384, SHA-512*], key size [assignment: *key size (in bits) used in HMAC*], and message digest sizes [selection: *160, 224, 256, 384, 512*] bits that meet the following: FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code, and FIPS PUB 180-3, “Secure Hash Standard.”

¶ 1281 **Assurance Activity:**

¶ 1282 ***Test:***

¶ 1283 The evaluator shall use "The Keyed-Hash Message Authentication Code (HMAC) Validation System (HMACVS)" as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of

the algorithms known to be good that can produce test vectors that are verifiable during the test.

D.2.6 FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition

(selected with FCS_IPSEC_EXT.1.4)

Hierarchical to: No other components.

Dependencies: FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

¶ 1284 ***Application Note:***

¶ 1285 *The TOE must support pre-shared keys for use in the IPsec protocol. There are two types of pre-shared keys--text-based (which are required) and bit-based (which are optional)--supported by the TOE, as specified in the requirements below. The first type is referred to as "text-based pre-shared keys", which refer to pre-shared keys that are entered by users as a string of characters from a standard character set, similar to a password. Such pre-shared keys must be conditioned so that the string of characters is transformed into a string of bits, which is then used as the key.*

¶ 1286 *The second type is referred to as "bit-based pre-shared keys" (for lack of a standard term); this refers to keys that are either generated by the TSF on a command from the administrator, or input in "direct form" by an administrator. "Direct form" means that the input is used directly as the key, with no "conditioning" as was the case for text-based pre-shared keys. An example would be a string of hex digits that represent the bits that comprise the key.*

¶ 1287 *The requirements below mandate that the TOE must support text-based pre-shared keys and optionally support bit-based pre-shared keys, although generation of the bit-based pre-shared keys may be done either by the TOE or in the Operational Environment.*

¶ 1288 **FIA_PSK_EXT.1.1** The TSF shall be able to use pre-shared keys for IPsec.

¶ 1289 **FIA_PSK_EXT.1.2** The TSF shall be able to accept text-based pre-shared keys that are:

- 22 characters in length and [selection: [assignment: *other supported lengths]*, *no other lengths*];

- composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”).

¶ 1290 **FIA_PSK_EXT.1.3** The TSF shall condition the text-based pre-shared keys by using [selection: *SHA-1*, *SHA-256*, *SHA-512*, [assignment: *method of conditioning text string*]] and be able to [selection: *use no other pre-shared keys; accept bit-based pre-shared keys; generate bit-based pre-shared keys using the random bit generator specified in FCS_RBG_EXT.1*].

¶ 1291 ***Application Note:***

¶ 1292 *For the length of the text-based pre-shared keys, a common length (22 characters) is required to help promote interoperability. If other lengths are supported they should be listed in the assignment; this assignment can also specify a range of values (e.g., "lengths from 5 to 55 characters") as well.*

¶ 1293 *In the second selection for FIA_PSK_EXT.1.3, the ST author fills in the method by which the text string entered by the administrator is “conditioned” into the bit string used as the key. This can be done by using one of the specified hash functions, or some other method through the assignment statement. If “bit-based pre-shared keys” is selected, the ST author specifies whether the TSF merely accepts bit-based pre-shared keys, or is capable of generating them. If it generates them, the requirement specified that they must be generated using the RBG specified by the requirements. If the use of bit-based pre-shared keys is not supported, the ST author chooses “use no other pre-shared keys”.*

¶ 1294 **Assurance Activity:**

¶ 1295 ***Operational Guidance:***

¶ 1296 The evaluator shall examine the operational guidance to determine that it provides guidance on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2.

¶ 1297 ***TSS:***

¶ 1298 The evaluator shall examine the TSS to ensure that it states that text-based pre-

shared keys of 22 characters are supported, and that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key sequence entered by the user (e.g., ASCII representation) to the bit string used by IPsec, and that this conditioning is consistent with the first selection in the FIA_PSK_EXT.1.3 requirement. If the assignment is used to specify conditioning, the evaluator will confirm that the TSS describes this conditioning.

¶ 1299 If "bit-based pre-shared keys" is selected, the evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall also examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1.

¶ 1300 ***Test:***

¶ 1301 The evaluator shall also perform the following tests:

1. The evaluator shall compose at least 15 pre-shared keys of 22 characters that cover all allowed characters in various combinations that conform to the operational guidance, and demonstrates that a successful protocol negotiation can be performed with each key.
2. [conditional]: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length; and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE.
3. [conditional]: If the TOE supports bit-based pre-shared keys but does not generate such keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.
4. [conditional]: If the TOE supports bit-based pre-shared keys and does generate such keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a

successful protocol negotiation can be performed with the key.

D.3 Trusted Update

D.3.1 FCS_COP.1(c) Cryptographic operation (Hash Algorithm)

(selected in FPT_TUD_EXT.1.3, or with FCS_SNI_EXT.1.1)

Hierarchical to: No other components.

Dependencies: No dependencies.

¶ 1302 **FCS_COP.1.1(c) Refinement**: The TSF shall perform **cryptographic hashing services** in accordance with [selection: ***SHA-1, SHA-256, SHA-384, SHA-512***] that meet the following: [**ISO/IEC 10118-3:2004**].

¶ 1303 ***Application Note (for O.STORAGE_ENCRYPTION):***

¶ 1304 *The hash selection should be consistent with the overall strength of the algorithm used for FCS_COP.1(d). (SHA 256 should be chosen for AES 128-bit keys, SHA 512 should be chosen for AES-256-bit keys) The selection of the standard is made based on the algorithms selected.*

¶ 1305 *Vendors are strongly encouraged to implement updated protocols that support the SHA-2 family; until updated protocols are supported, this PP allows support for SHA-1 implementations in compliance with SP 800-131A.*

¶ 1306 **Assurance activity:**

¶ 1307 ***TSS:***

¶ 1308 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.

¶ 1309 ***Operational Guidance:***

¶ 1310 The evaluator checks the operational guidance documents to determine that any configuration that is required to be done to configure the functionality for the required hash sizes is present.

¶ 1311 ***Test:***

¶ 1312 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 test mode.

¶ 1313 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.

¶ 1314 Short Messages Test - Bit-oriented Mode

¶ 1315 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 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.

¶ 1316 Short Messages Test - Byte-oriented Mode

¶ 1317 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 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.

¶ 1318 Selected Long Messages Test - Bit-oriented Mode

¶ 1319 The evaluators devise an input set consisting of m messages, where m is the block length of the hash algorithm. For SHA-256, the length of the i-th message is $512 + 99*i$, where $1 \leq i \leq m$. For SHA-512, the length of the i-th message is $1024 + 99*i$, 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.

¶ 1320 Selected Long Messages Test - Byte-oriented Mode

¶ 1321 The evaluators devise an input set consisting of m/8 messages, where m is the block length of the hash algorithm. For SHA-256, the length of the i-th message is 512 + 8*99*i, where 1 ≤ i ≤ m/8. For SHA-512, the length of the i-th message is 1024 + 8*99*i, where 1 ≤ i ≤ m/8. 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.

¶ 1322 Pseudorandomly Generated Messages Test

¶ 1323 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 The Secure Hash Algorithm Validation System (SHAVS). The evaluators then ensure that the correct result is produced when the messages are provided to the TSF.

D.4 Passphrase-based Key Entry

¶ 1324 The SFRs in this section are to be incorporated in the ST to support the optional Passphrase-based Key Entry function.

D.4.1 FCS_PCC_EXT.1 Extended: Cryptographic Password Construct and Conditioning

(for O. STORAGE_ENCRYPTION)

Hierarchical to: No other components

Dependencies: FCS_COP.1(h) Cryptographic Operation (for keyed-hash message authentication)

¶ 1325 **FCS_PCC_EXT.1.1** A password used to generate a password authorization factor shall enable up to [assignment: *positive integer of 64 or more*] characters in the set of {upper case characters, lower case characters, numbers, and [assignment: *other supported special characters*]} and shall perform Password-based Key Derivation Functions in accordance with a specified cryptographic algorithm [HMAC-[selection: *SHA-256, SHA-384, SHA-512*]], with [assignment: *positive integer of 1000 or more*] iterations, and output cryptographic key sizes [selection: *128, 256*] that meet the following: [**NIST SP 800-132**].

¶ 1326 ***Application Note:***

¶ 1327 *This SFR is conditionally required if the manual entry of a drive encryption passphrase is supported by the TOE.*

¶ 1328 **Assurance activity:**

¶ 1329 ***TSS:***

¶ 1330 The evaluator shall ensure the TSS describes the manner in which the TOE enforces the construction of passwords, including the length, and requirements on characters (number and type). The TSS also provides a description of how the password is conditioned and the evaluator ensures it satisfies the requirement.

¶ 1331 ***KMD:***

¶ 1332 The evaluator shall examine the KMD to ensure that the formation of the BEV and intermediary keys is described and that the key sizes match that selected by the ST Author.

¶ 1333 The evaluator shall check that the KMD describes the method by which the password/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 concerning the hash function itself. The evaluator shall verify that the KMD contains a description of how the output of the hash function is used to form the submask that will be input into the function and is the same length as the BEV as specified above.

¶ 1334 ***Test:***

¶ 1335 The evaluator shall also perform the following tests:

¶ 1336 Test 1: Ensure that the TOE supports passwords/passphrases of a minimum length of 64 characters.

¶ 1337 Test 2: If the TOE supports a password/passphrase length up to a maximum number of characters, n (which would be greater than 64), then ensure that the TOE will not accept more than n characters.

¶ 1338 Test 3: Ensure that the TOE supports passwords consisting of all characters assigned and supported by the ST author.

D.4.2 FCS_KDF_EXT Extended: Cryptographic Key Derivation

(for O. STORAGE_ENCRYPTION)

Hierarchical to: No other components

Dependencies: FCS_COP.1(h) Cryptographic Operation (for keyed-hash message authentication),

[if selected: FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)]

¶ 1339 **FCS_KDF_EXT.1.1** The TSF shall accept [selection: *a RNG generated submask as specified in FCS_RBG_EXT.1, a conditioned password submask, imported submask*] to derive an intermediate key, as defined in [selection: *NIST SP 800-108 [selection: KDF in Counter Mode, KDF in Feedback Mode, KDF in Double-Pipeline Iteration Mode], NIST SP 800-132*], using the keyed-hash functions specified in FCS_COP.1(h), such that the output is at least of equivalent security strength (in number of bits) to the BEV.

¶ 1340 **Assurance Activity:**

¶ 1341 ***TSS:***

¶ 1342 The evaluator shall verify the TSS includes a description of the key derivation function and shall verify the key derivation uses an approved derivation mode and key expansion algorithm according to SP 800-108 and SP800-132.

¶ 1343 ***KMD:***

¶ 1344 The evaluator shall examine the vendor's KMD to ensure that all keys used are derived using an approved method and a description of how and when the keys are derived.

D.4.3 FCS_COP.1(h) Cryptographic Operation (for keyed-hash message authentication)

(selected with FCS_PCC_EXT.1, FCS_KDF_EXT.1.1)

Hierarchical to: No other components.

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(b) Cryptographic key generation (Symmetric Keys)]

FCS_COP.1(c) Cryptographic operation (Hash Algorithm),

FCS_CKM_EXT.4 Extended: Cryptographic Key Material

Destruction

¶ 1345 **FCS_COP.1.1(h) Refinement:** The TSF shall perform [**keyed-hash message authentication**] in accordance with **[selection:** *HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-512***]** and cryptographic key sizes [assignment: *key size (in bits) used in HMAC*] that meet the following: [**ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"; ISO/IEC 10118**].

¶ 1346 ***Application Note:***

¶ 1347 *The key size [k] in the assignment falls into a range between L1 and L2 (defined in ISO/IEC 10118 for the appropriate hash function for example for SHA-256 L1 = 512, L2 =256) where L2 ≤ k ≤ L1.*

¶ 1348 **Assurance Activity:**

¶ 1349 ***TSS:***

¶ 1350 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.

¶ 1351 ***Test:***

¶ 1352 For each of the supported parameter sets, the evaluator shall compose 15 sets of test data. Each set shall consist of a key and message data. The evaluator shall have the TSF generate HMAC tags for these sets of test data. The resulting MAC tags shall be equal to the result of generating HMAC tags with the same key using a known good implementation.

D.4.4 FCS_SNI_EXT.1 Extended: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

(selected with FCS_PCC_EXT.1, FCS_KDF_EXT.1.1)

Hierarchical to: No other components

Dependencies: FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

¶ 1353 **FCS_SNI_EXT.1.1** The TSF shall only use salts that are generated by a **RNG as specified in FCS_RBG_EXT.1**.

¶ 1354 **FCS_SNI_EXT.1.2** The TSF shall only use unique nonces with a minimum size of [64]

bits.

¶ 1355 **FCS_SNI_EXT.1.3** The TSF shall create IVs in the following manner: [

- CBC: IVs shall be non-repeating,
- CCM: Nonce shall be non-repeating.
- XTS: No IV. Tweak values shall be non-negative integers, assigned consecutively, and starting at an arbitrary non-negative integer,
- GCM: IV shall be non-repeating. The number of invocations of GCM shall not exceed 2^32 for a given secret key.

¶ 1356].

¶ 1357 ***Application Note:***

¶ 1358 *This SFR is conditionally required if the manual entry of a drive encryption passphrase is supported by the TOE.*

¶ 1359 **Assurance activity:**

¶ 1360 ***TSS:***

¶ 1361 The evaluator shall ensure the TSS describes how salts are generated. The evaluator shall confirm that the salt is generating using an RBG described in FCS_RBG_EXT.1.

¶ 1362 The evaluator shall ensure the TSS describes how nonces are created uniquely and how IVs and tweaks are handled (based on the AES mode). The evaluator shall confirm that the nonces are unique and the IVs and tweaks meet the stated requirements.

Appendix E Entropy Documentation and Assessment

¶ 1363 This appendix describes the required supplementary information for each entropy source used by the TOE.

¶ 1364 The documentation of the entropy source(s) 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.

E.1 Design Description

¶ 1365 Documentation shall include the design of each 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.

¶ 1366 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.

¶ 1367 If implemented, the design description shall include a description of how third-party applications can add entropy to the RBG. A description of any RBG state saving between power-off and power-on shall be included.

E.2 Entropy Justification

¶ 1368 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.

¶ 1369 The entropy justification shall not include any data added from any third-party application or from any state saving between restarts.

E.3 Operating Conditions

¶ 1370 Documentation will also include the range of operating conditions under which the entropy source is expected to generate random data. Similarly, documentation shall describe the conditions under which the entropy source is no longer guaranteed to provide sufficient entropy. Methods used to detect failure or degradation of the source shall be included.

E.4 Health Testing

¶ 1371 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, TOE behavior upon entropy source failure, and rationale indicating why each test is believed to be appropriate for detecting one or more failures in the entropy source.

Appendix F Key Management Description

¶ 1372 The documentation of the product's encryption key management should be detailed enough that, after reading, the evaluator will thoroughly understand the product's key management and how it meets the requirements to ensure the keys are adequately protected. This documentation should include an essay and diagram(s). This documentation is not required to be part of the TSS - it can be submitted as a separate document and marked as developer proprietary.

F.1 Essay

¶ 1373 The essay will provide the following information for all keys in the key chain:

- The purpose of the key
- If the key is stored in nonvolatile memory
- How and when the key is protected
- How and when the key is derived
- The strength of the key
- When or if the key would be no longer needed, along with a justification.
- Key destruction description

¶ 1374 The essay will also describe the following topics:

¶ 1375 A description of all authorization factors that are supported by the product and how each factor is handled, including any conditioning and combining performed.

¶ 1376 If validation is supported, the process for validation shall be described, noting what value is used for validation and the process used to perform the validation. It shall describe how this process ensures no keys in the key chain are weakened or exposed by this process.

¶ 1377 The authorization process that leads to the ultimate release of the BEV. This section shall detail the key chain used by the product. It shall describe which keys are used in the protection of the BEV and how they meet the derivation, key wrap, or a combination of the two requirements, including the direct chain from the initial authorization to the BEV. It shall also include any values that add into that key chain or interact with the key chain and the protections that ensure those

values do not weaken or expose the overall strength of the key chain.

¶ 1378 The diagram and essay will clearly illustrate the key hierarchy to ensure that at no point the chain could be broken without cryptographically exhausting all of the initial authorization values and that the effective strength of the BEV is maintained throughout the Key Chain.

¶ 1379 A description of the data encryption engine, its components, and details about its implementation (e.g. for hardware: integrated within the device's main SOC or separate co-processor, for software: initialization of the product, drivers, libraries (if applicable), logical interfaces for encryption/decryption, and areas which are not encrypted (e.g. boot loaders, portions associated with the Master Boot Record (MBRs), partition tables, etc.)). The description should also include the data flow from the device's host interface to the device's persistent media storing the data, information on those conditions in which the data bypasses the data encryption engine (e.g. read-write operations to an unencrypted Master Boot Record area). The description should be detailed enough to verify all platforms to ensure that when the user enables encryption, the product encrypts all hard storage devices. It should also describe the platform's boot initialization, the encryption initialization process, and at what moment the product enables the encryption.

¶ 1380 The process for destroying keys when they are no longer needed by describing the storage location of all keys and the protection of all keys stored in nonvolatile memory.

F.2 Diagram

¶ 1381 The diagram will include all of keys from the initial authorization factor(s) to the BEV and any keys or values that contribute into the chain. It must list the cryptographic strength of each key and indicate how each key along the chain is protected with either Key Derivation or Key Wrapping (from the allowed options). The diagram should indicate the input used to derive or unwrap each key in the chain.

¶ 1382 A functional (block) diagram showing the main components (such as memories and processors) and the data path between, for hardware, the device's host interface and the device's persistent media storing the data, or for software, the initial steps needed to the activities the TOE performs to ensure it encrypts the storage device entirely when a user or administrator first provisions the product. The hardware encryption diagram shall

show the location of the data encryption engine within the data path.

¶ *1383* The evaluator shall validate that the hardware encryption diagram contains enough detail showing the main components within the data path and that it clearly identifies the data encryption engine.

Appendix G Terminology

Table 18 Glossary

Term	Definition	Source
Address Book	Electronic storage mechanism that equates names of persons or physical locations with machine-usable destinations (e.g., fax telephone numbers, email addresses, Uniform Resource Locators).	
Administrator	A User who has been specifically granted the authority to manage some portion or all of the TOE and whose actions may affect the security policies of the TOE. Administrators may possess special privileges that provide capabilities to override portions of security policies.	[2600.1]
Asset	Entities that the owner of the TOE presumably places value upon.	[CC]
Assumption	Physical, technical, and administrative conditions or requirements of the Operational Environment that must be upheld in order for the TOE to provide security functionality.	
Commercial Off-The-Shelf	Products that are both commercial and sold in substantial quantities in the commercial marketplace, and that can be procured or utilized under government contract in the same precise form as available to the general public.	[FAR]
Conditionally Mandatory Uses	One of the uses described in section 1.3.1.2 which, if present in the TOE, must be included in its evaluated configuration.	

Term	Definition	Source
Confidential (TSF) Data	Assets for which either disclosure or alteration by a User who is not an Administrator or the owner of the data would have an effect on the operational security of the TOE.	[2600.1]
Create	Assigning a value or content to data in a storage device. Note that in the case of document processing jobs, the outcome is that the job is initiated	
Credentials	A form of authentication data that specifies basic identifying information about a User or application. Credentials may be bound in some way to the individual to whom they were issued, or they may be bearer Credentials. The former are necessary for identification, while the latter may be acceptable for some forms of authorization.	[2600]
Decommission	The act of retiring an HCD from active use in the Operational Environment. It may also involve a change in geographic location and/or ownership.	
Delete	Dereferencing or otherwise making unavailable data in a storage device. Note that in the case of document processing jobs, the outcome is that the job is terminated.	
Document	A medium and the information recorded on it that generally has permanence and can be read by a person or a machine.	[610.12]
Document Processing	Printing, scanning, or copying a Document.	

Term	Definition	Source
Document Processing Job	A User request to the TOE to perform a Document Processing operation on a Document.	
External Authentication	Identification and authentication mechanism that uses services of External IT Entities to authenticate TOE Users.	
External IT Entity	An External Entity that is an IT device (not a human).	[CC] defines "External Entity"
Field-Replaceable (Unit)	The smallest subassembly that can be swapped in the field to repair a fault.	[IEEE]
Hardcopy Device	A system producing or utilizing a physical embodiment of an electronic document or image. These systems include printers, scanners, fax machines, digital copiers, MFPs (multifunction peripherals), MFDs (multifunction devices), "all-in-ones" and other similar products.	[2600]
Internal Authentication	Identification and authentication function that is wholly contained within the TOE.	
Job	A document processing task submitted to the hardcopy device. A single processing task may process one or more documents.	[2600.1]
Job Owner	A User who has permission to control a Job and access its documents. Typically, such permissions are obtained by submitting a Job, by access control mechanism, or by obtaining a credential associated with a Job.	

Term	Definition	Source
Local Area Network	A non-public data network in which serial transmission is used without store and forward techniques for direct data communication among data stations located on the User's premises.	[8802-6]
Local User	A User who is physically interacting with the HCD.	
Modify	Changing the value / content of data in a storage device. Note that in the case of document processing jobs, the outcome is that the instructions or other parameters of the job are changed.	
Multifunction Device	A Hardcopy Device that fulfills multiple purposes by using multiple functions in different combinations to replace several, single function devices. [Also known as Multifunction Printer and Multifunction Peripheral]	[2600]
Network Printing	Printing operation that has been initiated by a Network User.	
Network User	A User who interacts with the HCD over a network.	
Nonvolatile Storage Device	A device that provides computer storage of data that is not cleared when the power is turned off.	
Normal User	A User who is authorized to perform functions that process User Document Data in the TOE.	
Operational Environment	Environment in which the TOE is operated.	[CC]
Optional Use	One of the uses described in section 1.3.1.3 which may be present in the TOE, and may optionally be included in its evaluated configuration.	

Term	Definition	Source
Organizational Security Policy	Set of security rules, procedures, or guidelines for an organization.	[CC]
Output Tray	A receptacle for the TOE's printed output.	
Protected (TSF) Data	Assets for which alteration by a User who is not an Administrator or the owner of the data would have an effect on the operational security of the TOE, but for which disclosure is acceptable.	[2600.1]
Protection Profile	Implementation-independent statement of security needs for a TOE type.	[CC]
Read	To access data from a storage device or data medium. (Note that in this case, the data medium may be a printed output, and therefore, release of a print job is a “read” operation)	[610.12]
Redeploy	The act of moving an HCD from one Operational Environment to another Operational Environment.	
Required Use	One of the uses described in section 1.3.1.1 which must be present in the TOE in its evaluated configuration.	
Security Assurance Requirement	A description of how assurance is to be gained that the TOE meets the SFRs.	[CC]
Security Functional Requirement	A translation of the Security Objectives for the TOE into a standardized language.	[CC]
Security Objective	Statement of an intent to counter identified Threats and/or satisfy identified organization security policies and/or Assumptions.	[CC]

Term	Definition	Source
Security Target	Implementation-dependent statement of security needs for a specific identified TOE.	[CC]
Servicing	Performing repairs or preventative maintenance on the HCD.	
Standard Protection Profile	A Protection Profile that is developed according to processes defined by NIAP.	
Target of Evaluation	Set of software, firmware and/or hardware possibly accompanied by guidance.	[CC]
Temporary Storage	Storage of data that is not intentionally retained by the TOE after the completion of a Document Processing Job.	
Threat	Capabilities, intentions, and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.	[2600.1]
TOE Owner	A person or organizational entity responsible for protecting TOE Assets and establishing related security policies.	[2600.1]
TOE Security Functionality	Combined functionality of all hardware, software, and firmware of a TOE that must be relied upon for the correct enforcement of the SFRs.	[CC]
TSF Data	Data for the operation of the TOE upon which the enforcement of the SFR relies.	[CC]
TSF interface	Means by which external entities (or subjects in the TOE but outside of the TSF) supply data to the TSF, receive data from the TSF and invoke services from the TSF	[CC]

Term	Definition	Source
Unauthorized Access	Access to a resource that a User is not permitted to access.	
User	Human or IT entity possibly interacting with the TOE from outside of the TOE boundary.	[CC]
User Data	Data for the User that does not affect the operation of the TSF.	[CC]
User Document Data	The Asset that consists of the information contained in a User's Document. This includes the original Document itself in either hardcopy or electronic form, image data, or residually stored data created by the hardcopy device while processing an original Document and printed hardcopy output	[2600.1]
User Job Data	The Asset that consists of the information about a User's Document or job to be processed by the TOE.	[2600.1]

Sources:

[2600] IEEE Std. 2600™-2008 "IEEE Standard for Information Technology: Hardcopy Device and System Security"

[2600.1] IEEE Std. 2600.1™-2009 "IEEE Standard for a Protection Profile in Operational Environment A"

[610.12] IEEE Std 610.12-1990 "IEEE Standard Glossary of Software Engineering Terminology"

[8802-6] ISO /IEC 8802-6:1994 "Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements – Part 6"

[CC] ISO/IEC 15408-1:2009 "Information technology – Security techniques – Evaluation

criteria for IT security – Part 1"

[FAR] United States Federal Acquisition Regulations

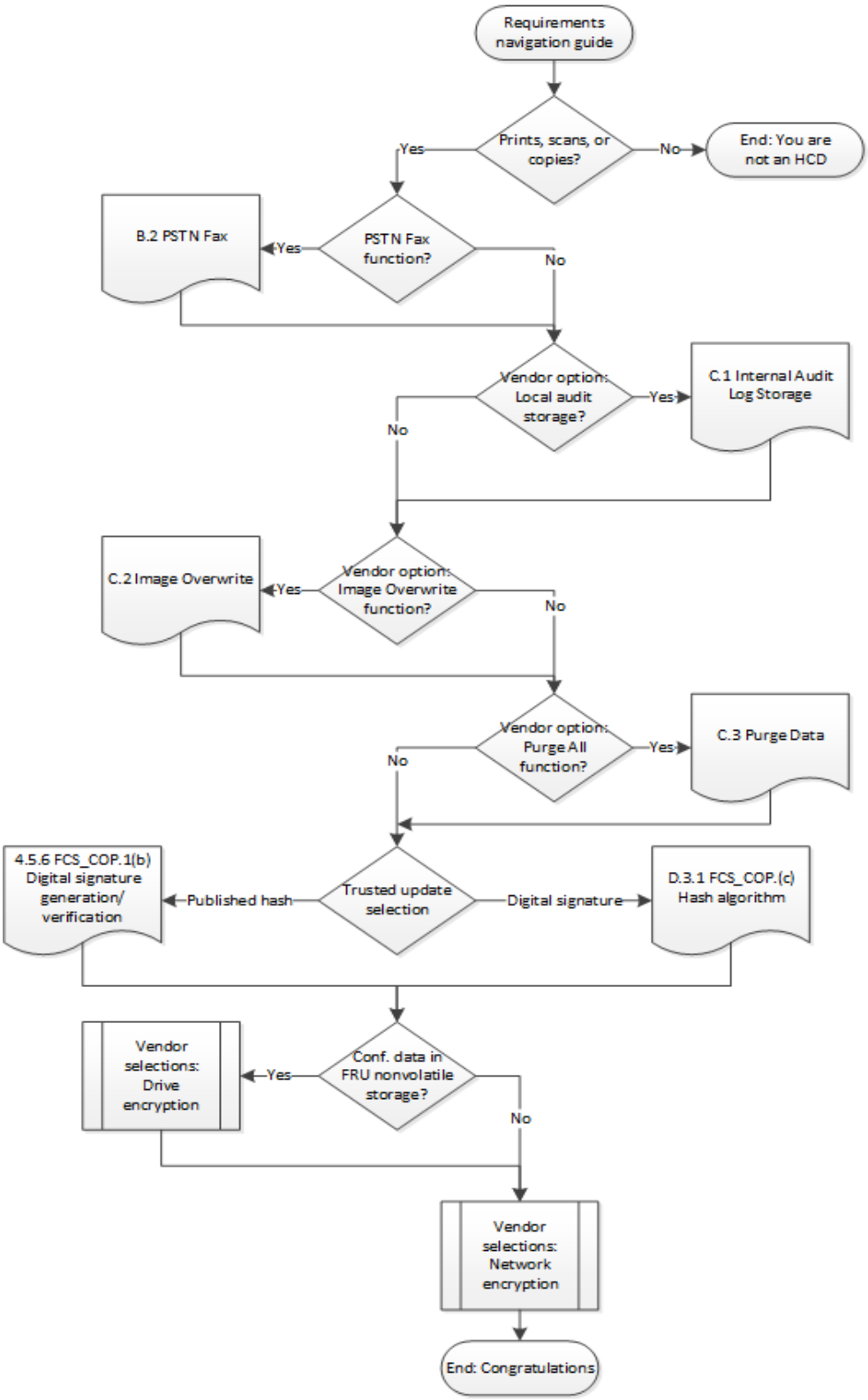
[IEEE] IEEE Standards Dictionary (ISBN 973-0-7381-2601-2)

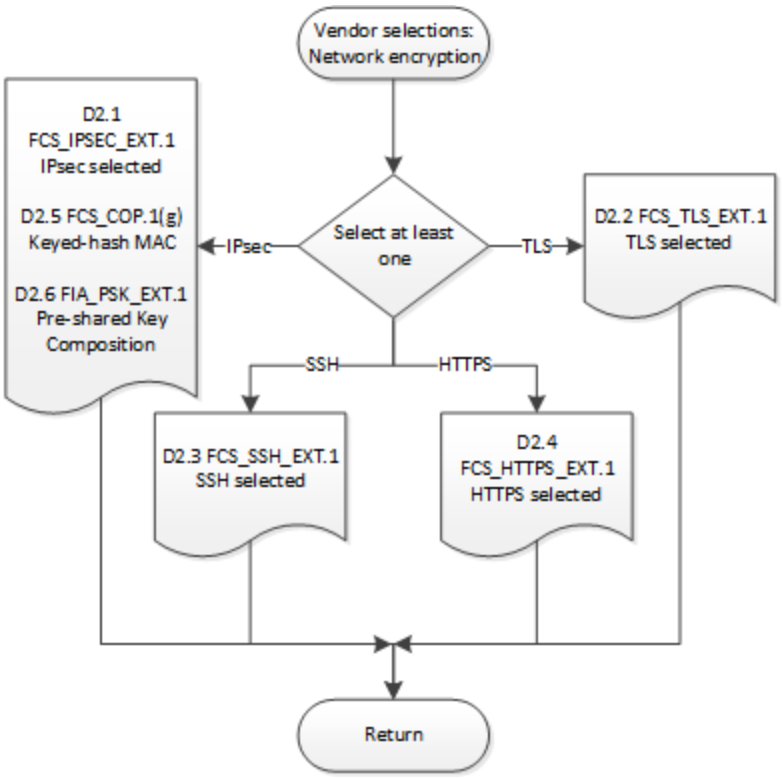
Table 19 Acronyms

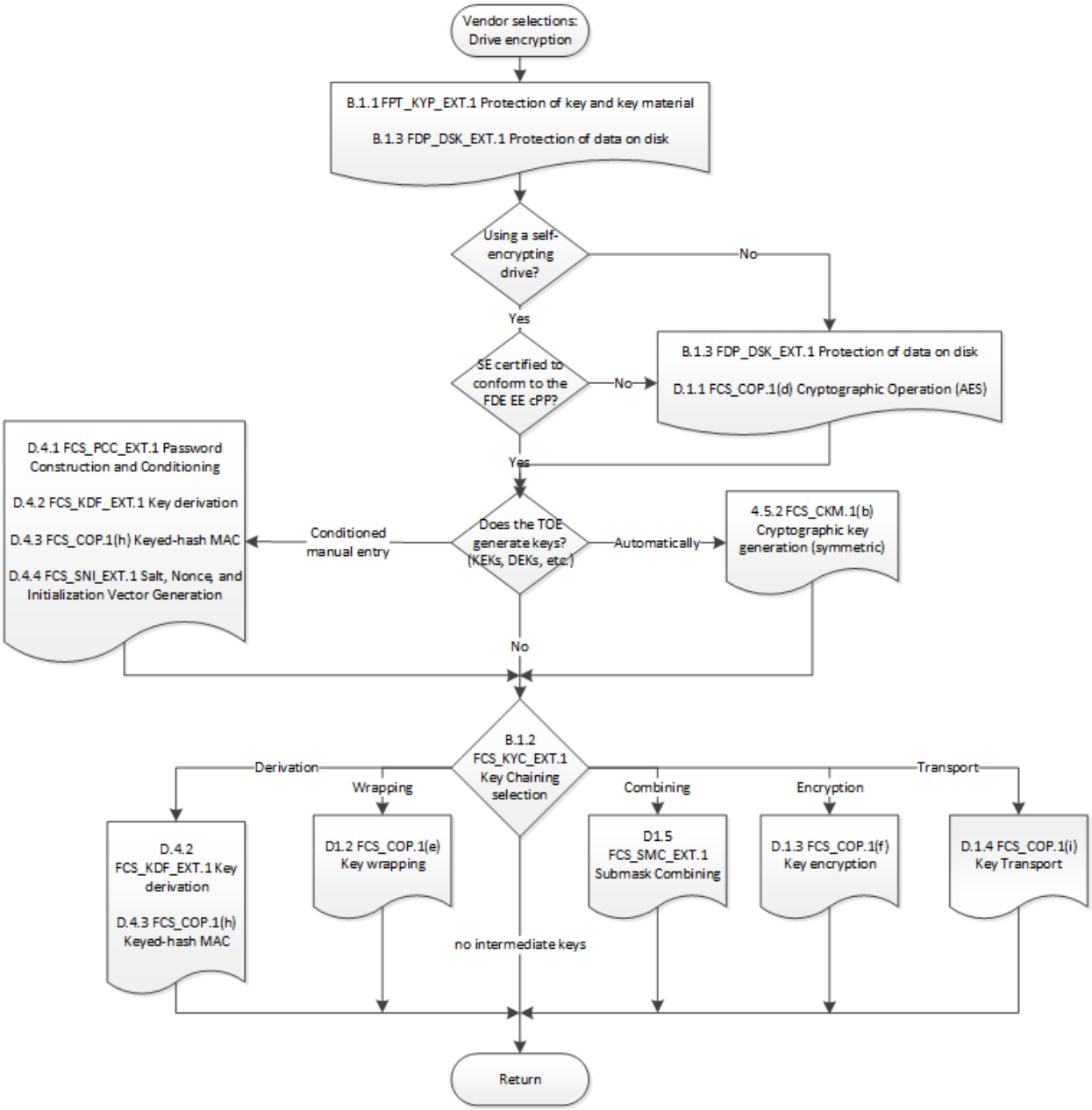
Acronym	Definition
BEV	Border Encryption Value
CC	Common Criteria
CCEVS	Common Criteria Evaluation and Validation Service
COTS	Commercial Off-The-Shelf
EAL	Evaluation Assurance Level
HCD	Hardcopy Device
IPA	Information-technology Promotion Agency
I&A	Identification and Authentication
IT	Information Technology
JISEC	Japan Information technology Security Evaluation and Certification scheme
KDF	Key Derivation Function
KMD	Key Management Description
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
MFD	Multifunction Device

Acronym	Definition
MFP	Multifunction Printer, Multifunction Peripheral
NIAP	National Information Assurance Partnership
OCSP	Online Certificate Status Protocol
OSP	Organizational Security Policy
PP	Protection Profile
PSTN	Public Switched Telephone Network
RBG	Random Bit Generator
SAR	Security Assurance Requirement
SFP	Security Functional Policy
SFR	Security Functional Requirement
SPP	Standard Protection Profile
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSFI	TSF Interface
TSS	TOE Summary Specification

Appendix H Protection Profile Navigation Guide




Vendor selections: Network encryption
D2.1 FCS_IPSEC_EXT.1 IPsec selected
D2.5 FCS_COP.1(g) Keyed-hash MAC
D2.6 FIA_PSK_EXT.1 Pre-shared Key Composition
IPsec
Select at least one
TLS
D2.2 FCS_TLS_EXT.1 TLS selected
SSH
HTTPS
D2.3 FCS_SSH_EXT.1 SSH selected
D2.4 FCS_HTTPS_EXT.1 HTTPS selected
Return


Vendor selections: Drive encryption
B.1.1 FPT_KYP_EXT.1 Protection of key and key material
B.1.3 FDP_DSK_EXT.1 Protection of data on disk
Using a self-encrypting drive?
No
Yes
SE certified to conform to the FDE EE cPP?
No
B.1.3 FDP_DSK_EXT.1 Protection of data on disk
D.1.1 FCS_COP.1(d) Cryptographic Operation (AES)
Yes
D.4.1 FCS_PCC_EXT.1 Password Construction and Conditioning
D.4.2 FCS_KDF_EXT.1 Key derivation
D.4.3 FCS_COP.1(h) Keyed-hash MAC
D.4.4 FCS_SNI_EXT.1 Salt, Nonce, and Initialization Vector Generation
Conditioned manual entry
Does the TOE generate keys? (KEKs, DEKs, etc.)
Automatically
4.5.2 FCS_CKM.1(b) Cryptographic key generation (symmetric)
No
B.1.2 FCS_KYC_EXT.1 Key Chaining selection
Derivation
Wrapping
Combining
Encryption
Transport
D.4.2 FCS_KDF_EXT.1 Key derivation
D.4.3 FCS_COP.1(h) Keyed-hash MAC
D1.2 FCS_COP.1(e) Key wrapping
D1.5 FCS_SMC_EXT.1 Submask Combining
D.1.3 FCS_COP.1(f) Key encryption
D.1.4 FCS_COP.1(i) Key Transport
no intermediate keys
Return