# collaborative Protection Profile for Hardcopy Devices

## Acknowledgements

This collaborative Protection Profile (cPP) was developed by the Hardcopy Device international Technical Community (iTC) also known as HCD-iTC with representatives from industry, Government agencies, Common Criteria Test Laboratories, and members of academia.

**INDUSTRY**

*Vendors*

**COMMON CRITERIA TESTING LABORATORIES**

*ITSEF Name*

**GOVERNMENT AGENCIES**

*Government Agency Names*

## Revision History

*Table 1. Revision history*

| Version | Date | Description |
|---|---|---|
| 0.6 | 8 June 2020 | Initial Release for HCD iTC Review |
| 0.7? | 21 July 2020? | Draft |
| 0.8 | 9 June 2021 | Draft |
| 0.9 | 16 August 2021 | Draft |
| 0.10 | 30 August 2021 | Public Draft 1 |
| 0.11 | 14 December 2021 | Public Draft 2 |
| 0.12.1 | 16 May 2022 | Public Draft Final |

## Table of Contents

# Preface

## Objectives of Document

This document presents the Common Criteria (CC) collaborative Protection Profile (cPP) to express the security functional requirements (SFRs) and security assurance requirements (SARs) for a Hardcopy Device (HCD). The Evaluation activities that specify the actions the evaluator performs to determine if a product satisfies the SFRs captured within this cPP, are described in [SD].

## Scope of Document

The scope of the cPP within the development and evaluation process is described in the Common Criteria for Information Technology Security Evaluation. In particular, a cPP defines the IT security requirements of a generic type of TOE and specifies the functional security measures to be offered by that TOE to meet stated requirements [[CC1], Section B.14].

## Intended Readership

The target audiences of this cPP are developers, CC consumers, system integrators, evaluators and schemes.

Although the cPP and SD may contain minor editorial errors, the cPP is recognized as living document and the iTC is dedicated to ongoing updates and revisions. Please report any issues to the HCD-iTC.

- [CC1] Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model, CCMB-2017-04-001, Version 3.1 Revision 5, April 2017.
- [CC2] Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components, CCMB-2017-04-002, Version 3.1 Revision 5, April 2017.
- [CC3] Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components, CCMB-2017-04-003, Version 3.1 Revision 5, April 2017.
- [CEM] Common Methodology for Information Technology Security Evaluation, Evaluation Methodology, CCMB-2017-04-004, Version 3.1 Revision 5, April 2017.
- [SD] Supporting Document Draft, v0.4, August 26, 2020 (https://ccusersforum.onlyoffice.com/Products/Files/doceditor.aspx?fileid=6744251&action=view)

For more see the Common Criteria Portal.

# 1. PP Introduction

## 1.1. PP Reference Identification

- PP Reference: collaborative Protection Profile for Hardcopy Devices
- PP Version: 0.12.1
- PP Date: 2022-05-16

## 1.2. TOE Overview

The Target of Evaluation in this cPP 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 Public Switched Telephone Network (PSTN) connection (PSTN faxing). Hardcopy documents typically take the form of paper, but can take other forms (e.g. transparencies).

For the purpose of this cPP, 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.

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” as well as "Optional 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. "Optional Uses" are optional functions that may, but need not, be evaluated.

# 1.3. TOE Design

## 1.3.1. Boundary of the TOE

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.

The logical boundary of the TOE includes all security functions related to the Required Uses of the HCD as described in Section 1.4.1, “USE CASE 1: Required Use Cases”, all Conditionally Mandatory Uses as described in Section 1.4.2, “USE CASE 2: Conditionally Mandatory Use Cases” that are present in the HCD, and all Optional Uses as described in Section 1.4.3, “USE CASE 3: Optional Use Cases” that are to be included in the evaluation.

## 1.3.2. Operational Environment

For the purposes of this cPP, 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.

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.

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

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. TOE Use Case

## 1.4.1. USE CASE 1: Required Use Cases

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

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.
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.
3. Auditing: Authorized personnel monitor security-relevant events in an audit log. The HCD generates audit log records when security-relevant events occur and stores them within the HCD. 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.
4. Verifying firmware/software updates: Authorized personnel install updated firmware/software on the HCD. The HCD ensures that only authorized personnel are permitted to install firmware/software, has the capability to help the installer to verify the authenticity of the firmware/software update.
5. Verifying HCD function: The HCD checks itself for malfunctions by performing a self-test and verifying firmware/software integrity each time that it is powered on.

## 1.4.2. USE CASE 2: Conditionally Mandatory Use Cases

Conditionally Mandatory Uses are security-relevant capabilities that are optional in the HCD, however if they are present, must be conforming:

1. 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.

2. 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.
3. 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.
4. 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 Nonvolatile Storage Devices from exposure if such a device is removed from the HCD. The HCD also has the capability to destroy cryptographic key so that the encrypted data cannot subsequently be decrypted.

### 1.4.3. USE CASE 3: Optional Use Cases

Optional Uses are security-relevant capabilities that are optional in the HCD, and even if present in the HCD, are not required to be evaluated:

1. 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.
2. 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 even if the data are encrypted and its cryptographic key is destroyed.

### 1.4.4. Major Security Functions of the HCD

To support the use cases in Section 1.4, "TOE Use Case", 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)

Each of these functions is described in the next subsections.

#### 1.4.4.1. Identification, Authentication, and Authorization

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.

#### 1.4.4.2. Access Control

Access controls ensure that Documents, information related to Document Processing, and security-relevant data are accessible only to Users who have appropriate access permissions.

#### 1.4.4.3. Data Encryption

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

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

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

#### 1.4.4.4. Trusted Communications

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

#### 1.4.4.5. Administrative Roles

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.

#### 1.4.4.6. Auditing

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 store them within the HCD as well as securely transmit them to an External IT entity for storage.

#### 1.4.4.7. Trusted Operation

Firmware/Software updates to the HCD are verified to ensure the authenticity of the firmware/software before applying the update. The HCD performs self-tests to ensure that its operation is not disrupted by some detectable malfunctions. The HCD performs hardware-anchored integrity verification of firmware/software at boot to ensure corrupted firmware/software is not

executed.

#### 1.4.4.8. PSTN Fax-Network Separation

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.

#### 1.4.4.9. Data Clearing and Purging

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, *Optional Requirements*.

# 2. CC Conformance Claims

As defined by the references [CC1], [CC2] and [CC3], this cPP:

- conforms to the requirements of Common Criteria v3.1, Revision 5,
- is Part 2 extended,
- is Part 3 conformant,
- does not claim conformance to any other security functional requirement packages.

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

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 1.4.1, “USE CASE 1: Required Use Cases”.
2. Security for all of those Required Uses supported by the TOE must be evaluated, conforming to the requirements of this Protection Profile.
3. If the TOE supports any of the Conditionally Mandatory Uses described in Section 1.4.2, “USE CASE 2: Conditionally Mandatory Use Cases”, then that support must be evaluated conforming to the corresponding conditionally mandatory requirements described in Appendix B, *Conditionally Mandatory Requirements*.
4. The selected communications protocol(s) must be evaluated conforming to the corresponding selection-based protocol requirements in Section A.2, “Protected Communications”.
5. The Security Target author may choose to include for evaluation any of the Optional Uses described in Section 1.4.3, “USE CASE 3: Optional Use Cases”. The vendor may choose to evaluate those optional functions as described in Appendix C, *Optional Requirements*.
6. The TOE must demonstrate Exact Conformance. Exact Conformance, as a subset of Strict Conformance as defined in [CC1], Annex D.2, 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.

# 3. Security Problem Definition

The security problem is described in terms of the threats that the TOE is expected to address, assumptions about its operational environment, and any organizational security policies that the TOE is expected to enforce.

## 3.1. Users

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.

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.

Note that a User can be a human user or an external IT entity. Also, a Normal User can be a Local User or a Network User as described in Section 1.3.2, “Operational Environment”.

Additional details about Users are in Section I.1, “User Definitions”.

## 3.2. Assets

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.

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.

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.

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].

Examples of assets requiring protection include transmitted communication data on the network

(against unauthorized disclosure or modification), firmware and/or software in the HCD (against unauthorized modification or deletion), and audit records generated by the HCD (against unauthorized modification or deletion).

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].

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 firmware/software of the TOE, and attacking External IT Entities.

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

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.

Additional details about assets are in Section I.2, "Asset Definitions".

## 3.3. Threats

The following are Threats against the TOE that are countered by conforming products. Additional details about threats are in Section I.3, "Threat Definitions".

### 3.3.1. Unauthorized Access to User Data

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.

### 3.3.2. Unauthorized Access to TSF Data

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.

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

### 3.3.3. Network Communication Attacks

An attacker may access data in transit or otherwise compromise the security of the TOE by monitoring or manipulating network communication [T.NET_COMPROMISE]. 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.

### 3.3.4. Malfunction

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 firmware/software malfunctions can produce unpredictable results, with a possibility that security functions will not operate correctly.

### 3.3.5. Weak Cryptography

An unauthorized user or attacker that observes network traffic transmitted to and from the TOE may cryptographically exploit poorly chosen cryptographic algorithms, random bit generators, ciphers or key sizes [T.WEAK_CRYPTO].

## 3.4. Assumptions

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 Section I.5, “Assumption Definitions”.

### 3.4.1. Physical Security

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.

### 3.4.2. Network Security

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.

### 3.4.3. Administrator Trust

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.

### 3.4.4. User Training

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.

## 3.5. Organizational Security Policies

The following are Organizational Security Policies (OSPs) that are upheld by conforming products. Additional details about OSPs are in Section I.4, “Organizational Security Policy Definitions”.

### 3.5.1. User Authorization

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.

### 3.5.2. Auditing

Security-relevant activities must be audited and the log of such actions must be stored within the TOE as well as protected and transmitted to an External IT Entity [P.AUDIT]. Stored internally as well as on an External IT Entity, 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.

### 3.5.3. Protected Communications

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.

### 3.5.4. Storage Encryption

If the TOE stores User Document Data or Confidential TSF Data on Nonvolatile Storage Devices, 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 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.

Cleartext keys, submasks, random numbers, or any other values that contribute to the creation of encryption keys for Nonvolatile Storage of User Document Data or Confidential TSF Data must be protected from unauthorized access and must not be stored on any Nonvolatile Storage Device without protection [P.KEY_MATERIAL]. Unauthorized possession of key material in cleartext may

allow an attacker to decrypt User Document Data or Confidential TSF Data.

### 3.5.5. PSTN Fax-Network Separation (conditionally mandatory)

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.

### 3.5.6. Image Overwrite (optional)

Upon completion or cancellation of a Document Processing job, periodically, or when requested by an authorized administrator, residual image data in the TOE shall be made irretrievable from its Nonvolatile Storage Devices [P.IMAGE_OVERWRITE]. A customer may be concerned that image data that has been dereferenced by the TOE operating firmware/software may remain on 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.

### 3.5.7. 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 [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 even if the data are encrypted and its cryptographic key is destroyed.

Note: Cryptographic erase is covered in the mandatory requirement of FCS_CKM_EXT.4 and FCS_CKM.4, so that it is not included in this optional requirement.

### 3.5.8. Root of Trust

The vendor provides a Root of Trust (RoT) that is comprised of the TOE firmware, hardware, and pre-installed public keys or required critical security parameters, free of intentionally malicious capabilities [P.ROT_INTEGRITY]. The platform trusts the RoT since it cannot verify the integrity and authenticity of the RoT.

# 4. Security Objectives

## 4.1. Security Objectives for the TOE

The following Security Objectives must be fulfilled by the TOE. Additional details about objectives

for the TOE are in Section I.6, “Definitions of Security Objectives for the TOE” and Section I.7, “Definitions of Security Objectives for the Operational Environment”.

### 4.1.1. User Authorization

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

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.

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.

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.

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.

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

### 4.1.2. User Identification and Authentication

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

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.

### 4.1.3. Access Control

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

The guiding principles for access control security policies in this cPP are:

1. User Document Data [D.USER.DOC] can be accessed only by the Document owner or an Administrator.
2. User Job Data [D.USER.JOB] can be read by any User but can be modified only by the Job Owner or an Administrator.
3. 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.

4. 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.

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

### 4.1.4. Administrator Roles

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

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.

### 4.1.5. Firmware/Software Update Verification

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

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

### 4.1.6. Self-test

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

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.

### 4.1.7. Communications 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.COMMS_PROTECTION]. This objective addresses the common concerns of network communications:

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

### 4.1.8. Auditing

The TOE shall generate audit data and store it internally as well as be capable of sending it to a trusted External IT Entity [O.AUDIT].

The TOE must store audit data internally with appropriate access controls to ensure confidentiality and integrity. Additionally, the TOE must be able to securely send audit data to a trusted External IT Entity (e.g., an audit server such as a syslog server).

### 4.1.9. Storage Encryption

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

This objective addresses the concern that User Document Data or Confidential TSF Data on a Nonvolatile Storage Device may be exposed if the device is removed from the TOE, such as for Servicing, Redeployment to another environment, or Decommissioning.

### 4.1.10. Protection of Key Material

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 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].

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.

### 4.1.11. PSTN Fax-Network 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.FAX_NET_SEPARATION].

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).

### 4.1.12. Image Overwrite (optional)

Upon completion or cancellation of a Document Processing job, periodically, or when requested by an authorized administrator, the TOE shall overwrite residual image data from its Nonvolatile Storage Devices [O.IMAGE_OVERWRITE]. This objective addresses customer concerns that image data may remain on Nonvolatile Storage Devices in the TOE after a Document Processing job has been completed or cancelled.

### 4.1.13. 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 [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.

### 4.1.14. Authentication Failures (conditionally mandatory)

The TOE resists repeated attempts to guess authorization data [O.AUTH_FAILURES] by responding to consecutive failed attempts in a way that prevents an attacker from exploring a significant amount of the space of possible authorization data values.

Note: This Security Objective needs to be Conditionally Mandatory based on the condition that the TOE has an internal authentication mechanism. Also, the HCD must ensure the HCD does not outlaw 3rd Party external authentication mechanisms.

### 4.1.15. Firmware Integrity

The TOE ensures its own integrity has remained intact [O.FW_INTEGRITY] and attests its integrity to outside parties on request.

### 4.1.16. Strong Cryptography

The TOE implements strong cryptographic mechanisms and algorithms according to recognized standards [O.STRONG_CRYPTO], including support for random bit generation based on recognized standards and a source of sufficient entropy. The TOE uses key sizes that are recognized as providing sufficient resistance to current attack capabilities.

## 4.2. Security Objectives for the Operational Environment

The following Security Objectives must be provided by the Operational Environment. Additional details about objectives for the Operational Environment are in Section I.7, “Definitions of Security Objectives for the Operational Environment”.

### 4.2.1. Physical Protection

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

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.

### 4.2.2. Network Protection

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

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.

### 4.2.3. Trusted Administrators

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

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.

### 4.2.4. Trained Users

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

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.

### 4.2.5. Trained Administrators

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].

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.

## 4.3. Security Objectives Rationale

The following table describes how the assumptions, threats, and organizational security policies map to the security objectives.

*Table 2. Mapping between Security Problem Defintion and Security Objectives*

| Security Objectives | Threat, Assumption, or OSP | Rationale |
|---|---|---|
| O.USER_I&A | T.UNAUTHORIZED_ACCESS<br>T.TSF_COMPROMISE<br>P.AUTHORIZATION | The TOE shall perform identification and authentication of Users for operations that require access control, User authorization, or Administrator roles. |
| O.ACCESS_CONTROL | T.UNAUTHORIZED_ACCESS<br>T.TSF_COMPROMISE<br>P.AUDIT | The TOE shall enforce access controls to protect User Data and TSF Data in accordance with security policies. |
| O.USER_AUTHORIZATION | P.AUTHORIZATION<br>P.AUDIT | The TOE shall perform authorization of Users in accordance with security policies. |
| O.ADMIN_ROLES | T.UNAUTHORIZED_ACCESS<br>T.TSF_COMPROMISE<br>P.AUTHORIZATION | The TOE shall ensure that only authorized Administrators are permitted to perform administrator functions. |
| O.UPDATE_VERIFICATION | T.UNAUTHORIZED_UPDATE | The TOE shall provide mechanisms to verify the authenticity of firmware/software updates. |
| O.TSF_SELF_TEST | T.TSF_FAILURE | The TOE shall test some subset of its security functionality to help ensure that subset is operating properly. |
| O.COMMS_PROTECTION | T.NET_COMPROMISE<br>P.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 | P.AUDIT | The TOE shall generate audit data and store it internally as well as be capable of sending it to a trusted External IT Entity. |
| O.STORAGE_ENCRYPTION | P.STORAGE_ENCRYPTION | If the TOE stores User Document Data or Confidential TSF Data in Nonvolatile Storage devices, then the TOE shall encrypt such data on those devices. |
| O.KEY_MATERIAL | P.KEY_MATERIAL | 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 Nonvolatile Storage Devices; The TOE shall ensure that such key material is not stored in cleartext on the storage device that uses that material. |

| Security Objectives | Threat, Assumption, or OSP | Rationale |
|---|---|---|
| O.FAX_NET_SEPARATION | P.FAX_FLOW | 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 | P.IMAGE_OVERWRITE | Upon completion or cancellation of a Document Processing job, the TOE shall overwrite residual image data from its Nonvolatile Storage Devices. |
| O.PURGE_DATA | P.PURGE_DATA | 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.AUTH_FAILURES | P.AUTHORIZATION | The TOE resists repeated attempts to guess authorization data by responding to consecutive failed attempts in a way that prevents an attacker from exploring a significant amount of the space of possible authorization data values. |
| O.FW_INTEGRITY | P.ROT_INTEGRITY | The TOE ensures its own integrity has remained intact and attests its integrity to outside parties on request. |
| O.STRONG_CRYPTO | T.WEAK_CRYPTO | The TOE implements strong cryptographic mechanisms and algorithms according to recognized standards, including support for random bit generation based on recognized standards and a source of sufficient entropy. The TOE uses key sizes that are recognized as providing sufficient resistance to current attack capabilities. |
| OE.PHYSICAL_PROTECTION | A.PHYSICAL | The Operational Environment shall provide physical security, commensurate with the value of the TOE and the data it stores or processes. |
| OE.NETWORK_PROTECTION | A.NETWORK | The Operational Environment shall provide network security to protect the TOE from direct, public access to its LAN interface. |
| OE.ADMIN_TRUST | A.TRUSTED_ADMIN | The TOE Owner shall establish trust that Administrators will not use their privileges for malicious purposes. |
| OE.USER_TRAINING | A.TRAINED_USERS | The TOE Owner shall ensure that Users are aware of site security policies and have the competence to follow them. |
| OE.ADMIN_TRAINING | A.TRAINED_USERS | 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. |

# 5. Security Functional Requirements

## 5.1. Conventions

The individual security functional requirements are specified in the sections below. The following conventions are used for the completion of operations:

- [*Italicized text within square brackets*] indicates an operation to be completed by the ST author.
- **Bold text** indicates additional text provided as a refinement.
- [**Bold text within square brackets**] indicates the completion of an assignment.
- [text within square brackets] indicates the completion of a selection.
- Iteration: indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").
- Extended SFRs are identified by having a label "EXT" at the end of the SFR name.

## 5.2. Security Audit (FAU)

### 5.2.1. FAU_GEN.1 Audit data generation

(for O.AUDIT) **Hierarchical to:**
No other components.

**Dependencies:**
FPT_STM.1 Reliable time stamps

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

a. Start-up and shutdown of the audit functions;
b. All auditable events for the **not specified** level of audit; and
c. **All auditable events specified in Table 3**, [assignment: *other specifically defined auditable events*].

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

a. Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
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 3**, [assignment: *other audit relevant information*].

*Table 3. Auditable Events*

| Auditable Event | Relevant SFR | Additional Information |
|---|---|---|
| Job Completion | FDP_ACF.1 | Type of Job |

| | | |
|---|---|---|
| Unsuccessful login attempts limit is met or exceeded | FIA_AFL.1 | None |
| Unsuccessful User authentication | FIA_UAU.1 | Supplied User ID/Name and origin of the attempt (e.g., IP address) |
| Unsuccessful User identification | FIA_UID.1 | Supplied User ID/Name and origin of the attempt (e.g., IP address) |
| Use of management functions | FMT_SMF.1 | Function that is invoked by user |
| 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/Admin, FTP_TRP.1/NonAdmin | Reason for failure |

***Application Note:***

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

*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.*

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

### 5.2.2. 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

**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.

### 5.2.3. FAU_SAR.1 Audit review

(for O.AUDIT)
**Hierarchical to:**
No other components.

**Dependencies:**
FAU_GEN.1 Audit data generation

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

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

### 5.2.4. FAU_SAR.2 Restricted audit review

(for O.AUDIT)
**Hierarchical to:**
No other components.

**Dependencies:**
FAU_SAR.1 Audit review

**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.

### 5.2.5. FAU_STG.1 Protected audit trail storage

(for O.AUDIT)
**Hierarchical to:**
No other components.

**Dependencies:**
FAU_GEN.1 Audit data generation

**FAU_STG.1.1** The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

**FAU_STG.1.2** The TSF shall be able to **prevent** unauthorized modifications to the stored audit records in the audit trail.

### 5.2.6. 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

**FAU_STG.4.1 Refinement:** The TSF shall [selection, choose one of: ~~***"ignore audited events"***~~ ,*"prevent audited events, except those taken by the authorized 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.

### 5.2.7. 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.

**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.

## 5.3. Cryptograhic Support (FCS)

### 5.3.1. FCS_CKM.1/SKG Cryptographic Key Generation (Symmetric Keys)

(for O.COMMS_PROTECTION, O.STORAGE_ENCRYPTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
FCS_COP.1/DataEncryption Cryptographic Operation (Symmetric encryption/decryption)
FCS_COP.1/StorageEncryption Cryptographic Operation (Data Encryption/Decryption)
FCS_COP.1/KeyWrap Cryptographic Operation (Key Wrapping)
FCS_COP.1/KeyEnc Cryptographic operation (Key Encryption)
FCS_COP.1/KeyedHash Cryptographic Operation (for keyed-hash message authentication)
FCS_COP.1/CMAC Cryptographic Operation (for keyed-hash message authentication)]
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction
FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

**FCS_CKM.1.1/SKG 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 bits, 192 bits, 256 bits*] that meet the following: [selection: *ISO/IEC 18031:2011 (Clause 9) [DRBG], NIST SP 800-133 Rev.2 Section [selection: 6.1, 6.3]]*.**

***Application Note:***

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

### 5.3.2. FCS_CKM.2 Cryptographic Key Establishment (Refinement)

(for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**FCS_CKM.2.1** The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: [selection:

- *RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2";*
- *Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";*
- *Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";*
- *FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [selection: RFC 3526, RFC 7919].*

] ~~that meets the following: [assignment: list of standards]~~.

***Application Note:***

*This is a refinement of the SFR FCS_CKM.2 to deal with key establishment rather than key distribution.*

*The ST author selects all key establishment schemes used for the selected cryptographic protocols.*

*The elliptic curves used for the key establishment scheme correlate with the curves specified in FCS_CKM.1.1/AKG.*

*The domain parameters used for the finite field-based key establishment scheme are specified by the key generation according to FCS_CKM.1.1/AKG.*

### 5.3.3. 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/AKG Cryptographic Key Generation (Asymmetric Keys), or
FCS_CKM.1/SKG Cryptographic Key Generation (Symmetric Keys) ],
FCS_CKM.2 Cryptographic Key Establishment
FCS_CKM.4 Cryptographic key destruction

**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.

***Application Note:***

*"Cryptographic Critical Security Parameters" are defined in ISO/IEC 19790:2012 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".*

*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.*

### 5.3.4. 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/AKG Cryptographic Key Generation (Asymmetric Keys), or
FCS_CKM.1/SKG Cryptographic Key Generation (Symmetric Keys) ]

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

- For volatile memory, the destruction shall be executed by a [selection: single overwrite consisting of [selection: a pseudo-random pattern using the TSF's RBG, zeroes, ones, a new value of a key, [assignment: any value that does not contain any CSP]], removal of power to the memory, destruction of reference to the key directly followed by a request for garbage

collection or memory management];

- For non-volatile storage that consists of the invocation of an interface provided by the underlying platform that [selection:
  - logically addresses the storage location of the key and performs a [selection: [selection: single, [assignment: ST author defined multi-pass]] overwrite consisting of [selection: zeroes, ones, pseudo-random pattern, a new value of a key of the same size, [assignment: any value that does not contain any CSP]], block erase];
  - instructs the underlying platform to destroy the abstraction that represents the key ]

] that meets the following: [**selection: *no standard***].

***Application Note:***

*In the first selection, the ST Author is presented options for destroying disused cryptographic keys based on whether they are in volatile memory or non-volatile memory within the TOE.*

*The interface provided by the underlying platform referenced in the requirement could take different forms, which is application programming interface to an OS kernel, a flash driver, a protected storage device, etc. As an illustrative example, the protected storage device is capable of destroying keys/secrets in the protected storage device upon request that is authorized by the protected storage device. The level of detail to which the TOE has access will be reflected in the TSS section of the ST.*

*Examples of protected storage device include Secure Elements (SE), Trusted Platform Modules (TPM), Hardware Security Modules (HSM), Trusted Execution Environments (TEE), and Secure Enclave Processors (SEP) and so on.*

### 5.3.5. FCS_COP.1/DataEncryption Cryptographic Operation (Data Encryption/Decryption)

(for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**FCS_COP.1.1/DataEncryption** The TSF shall perform *encryption/decryption* in accordance with specified cryptographic algorithms [selection:

- *AES used in* [selection: *CBC, CTR, GCM*] *mode*,
- SEED operating in [selection:
  - CBC mode with non-repeating and unpredictable IVs,
  - CFB mode with non-repeating and unpredictable IVs,
  - OFB mode with unique IVs,

    - CTR mode with unique, incremental counter,
    - CCM mode with unpredictable, non-repeating nonce, minimum size of 64 bits,
    - GCM mode with non-repeating IVs],
- HIGHT operating in [selection:
    - CBC mode with non-repeating and unpredictable IVs,
    - CFB mode with non-repeating and unpredictable IVs,
    - OFB mode with unique IVs,
    - CTR mode with unique, incremental counter],
- LEA operating in [selection:
    - CBC mode with non-repeating and unpredictable IVs,
    - CFB mode with non-repeating and unpredictable IVs,
    - OFB mode with unique IVs,
    - CTR mode with unique, incremental counter,
    - CCM mode with unpredictable, non-repeating nonce, minimum size of 64 bits,
    - GCM mode with non-repeating IVs]]

and cryptographic key sizes [selection:

Case: AES algorithm

- [selection: 128 bits, 192 bits, 256 bits],

Case: SEED algorithm

- 128 bits,

Case: HIGHT algorithm

- 128 bits,

Case: LEA algorithm

- [selection: 128 bits, 192 bits, 256 bits]]

that meet the following [selection:

Case: AES algorithm

- *ISO 18033-3*, [selection: *CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772*],

Case: SEED algorithm

- ISO/IEC 18033-3:2010, Subclause 5.4 “SEED”
- [selection:

    - ISO/IEC 10116:2017, Clause 7 “CBC”,
    - ISO/IEC 10116:2017, Clause 8 “CFB”,
    - ISO/IEC 10116:2017, Clause 9 “OFB”,
    - ISO/IEC 10116:2017, Clause 10 “CTR”,
    - ISO/IEC 19772:2009, Clause 8 “CCM”,
    - ISO/IEC 19772:2009, Clause 11 “GCM” and NIST SP800-38D],

Case: HIGHT algorithm

- ISO/IEC 18033-3:2010, Subclause 4.5 “HIGHT”
- [selection:
    - ISO/IEC 10116:2017, Clause 7 “CBC”,
    - ISO/IEC 10116:2017, Clause 8 “CFB”,
    - ISO/IEC 10116:2017, Clause 9 “OFB”,
    - ISO/IEC 10116:2017, Clause 10 “CTR],

Case: LEA algorithm

- ISO/IEC 29192-2:2019, Subclause 6.3 “LEA”
- [selection:
    - ISO/IEC 10116:2017, Clause 7 “CBC”,
    - ISO/IEC 10116:2017, Clause 8 “CFB”,
    - ISO/IEC 10116:2017, Clause 9 “OFB”,
    - ISO/IEC 10116:2017, Clause 10 “CTR”,
    - ISO/IEC 19772:2009, Clause 8 “CCM”,
    - ISO/IEC 19772:2009, Clause 11 “GCM” and NIST SP800-38D]

]

***Application Note:***

*For the first selection of FCS_COP.1.1/DataEncryption, the ST author chooses the mode or modes in which AES, SEED, HIGHT, or LEA operates. For the second selection, the ST author chooses the key sizes that are supported by this functionality. The modes and key sizes selected here correspond to the cipher suite selections made in the trusted channel requirements.*

*If either SEED in GCM mode with non-repeating IVs or LEA in GCM mode with non-repeating IVs is selected, IV length must be equal to 96 bits, the deterministic IV construction method [SP800-38D, Section 8.2.1] must be used, and the MAC length t must be one of the values 96, 104, 112, 120, and 128 bits.*

### 5.3.6. FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

(for O.UPDATE_VERIFICATION, O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
FCS_CKM.1/AKG Cryptographic Key Generation (Asymmetric Keys),
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**FCS_COP.1.1/SigGen** The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [selection:

- *RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [assignment: 2048 bits or greater],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [assignment: 256 bits or greater],*
- *Korean Certificate-based Digital Signature Algorithm (KCDSA) using [selection: SHA-224, SHA-256] with key size of 2048 bits,*
- *Elliptic Curve KCDSA (EC-KCDSA) on [selection: NIST P-224, NIST P-256, NIST B-233, NIST B-283, NIST K-233, NIST K-283] using [selection: SHA-224, SHA-256] with key size of [selection: 224 bits, 256 bits]]*

that meet the following: [selection:

Case: RSA schemes

- *FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,*

Case: Elliptic Curve Digital Signature Algorithm schemes

- *FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D,*
- *Implementing "NIST curves" [selection: P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4*

Case: Korean Certificate-based Digital Signature Algorithm

- ISO/IEC 14888-3:2018 (Subclause 6.3), "Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms"
- ISO/IEC 10118-3:2018 (Clause 10, 14), "Hash-functions – Part 3: Dedicated hash-functions"

Case: Elliptic Curve Korean Certificate-based Digital Signature Algorithm

- ISO/IEC 14888-3:2018 (Subclause 6.7), "Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms"

- FIPS186-4 (Appendix D.1.2, D.1.3), “Digital Signature Standard”
- The TSF shall implement “NIST curves” [selection: P-224, P-256, B-233, B-283, K-233, K-283] (as defined in FIPS PUB 186-4, “Digital Signature Standard”)
- ISO/IEC 10118-3:2018 (Clause 10, 14), “Hash-functions – Part 3: Dedicated hash-functions”

].

***Application Note:***

*The ST Author chooses the algorithm(s) implemented to perform digital signatures. For the algorithm(s) chosen, the ST author makes the appropriate assignments/selections to specify the parameters that are implemented for that algorithm. The ST author ensures that the assignments and selections for this SFR include all the parameter values necessary for the cipher suites selected for the protocol SFRs (see Section A.2, “Protected Communications”) that are included in the ST. The ST Author checks for consistency of selections with other FCS requirements, especially when supporting elliptic curves.*

### 5.3.7. FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

(for O.UPDATE_VERIFICATION, O.FW_INTEGRITY, O.STORAGE_ENCRYPTION, O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

**FCS_COP.1.1/Hash** The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [selection: *SHA-1, SHA-256, SHA-384, SHA-512*] ~~and cryptographic key sizes [assignment: cryptographic key sizes]~~ and **message digest sizes [selection: 160, 256, 384, 512] bits** that meet the following: *ISO/IEC 10118-3:2004*.

***Application Note:***

*Developers are strongly encouraged to implement updated protocols that support the SHA-2 family; until updated protocols are supported, this cPP allows support for SHA-1 implementations in compliance with SP 800-131A. In a future version of this cPP, SHA-256 will be the minimum requirement for all TOEs.*

*The hash selection should be consistent with the overall strength of the algorithm used for FCS_COP.1/DataEncryption and FCS_COP.1/SigGen (for example, SHA 256 for 128-bit keys).*

### 5.3.8. FCS_RBG_EXT.1 Random Bit Generation

(for O.STORAGE_ENCRYPTION, O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

**FCS_RBG_EXT.1.1** The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [selection: *Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG ([selection: AES, SEED, HIGHT, LEA])*].

**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 firmware/software-based sources*] firmware/software-based noise source, [*assignment: number of hardware-based sources*] hardware-based noise source] with a minimum of [selection: *128 bits, 192 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.

***Application Note:***

*For the first selection in FCS_RBG_EXT.1.2, the ST author selects at least one of the types of noise sources. If the TOE contains multiple noise sources of the same type, the ST author fills the assignment with the appropriate number for each type of source (e.g., 2 firmware/software-based noise sources, 1 hardware-based noise source). The documentation and tests required in the Evaluation Activity for this element should be repeated to cover each source indicated in the ST.*

*ISO/IEC 18031:2011 contains three 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, any of the block ciphers-based (AES, SEED, HIGHT, LEA) implementations for CTR_DRBG are allowed.*

*If the key length for the AES implementation used here is different than that used to encrypt the user data, then FCS_COP.1/DataEncryption may have to be adjusted or iterated to reflect the different key length. For the selection in FCS_RBG_EXT.1.2, the ST author selects the minimum number of bits of entropy that is used to seed the RBG, which must be equal or greater than the security strength of any key generated by the TOE.*

## 5.4. User Data Protection (FDP)

***Application Note:***

*The User Data Access Control SFP is composed of Table 4, Table 5, FDP_ACC.1, FDP_ACF.1, FMT_MSA.1, and FMT_MSA.3.*

### 5.4.1. 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

**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 4 and Table 5**.

### 5.4.2. 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

**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 4 and Table 5**.

**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 4 and Table 5***.

**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*].

**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 4. D.USER.DOC Access Control SFP*

| **PRINT** | "Create" | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| 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 |

| **SCAN** | "Create" | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| 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** | "Create" | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| 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** | "Create" | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| 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** | "Create" | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| 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** | "Create" | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| Operation: | Store document | Retrieve stored document | Modify stored document | Delete stored document |
| Job owner | (note 1) | | | |
| U.ADMIN | | | | |
| U.NORMAL | | denied | denied | denied |
| Unauthenticated | (condition 1) | denied | denied | denied |

*Table 5. D.USER.JOB Access Control SFP*

| "PRINT" | "Create" * | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| Operation: | Create print job | View print queue / job | Modify print job | Cancel print job |
| Job owner | (note 1) | | | |
| U.ADMIN | | | | |
| U.NORMAL | | | denied | denied |
| Unauthenticated | | | denied | denied |

| "SCAN" | "Create" * | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| 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" | "Create" * | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| 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" | "Create" * | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| 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" | "Create" * | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| 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" | "Create" * | "Read" | "Modify" | "Delete" |
|---|---|---|---|---|
| Operation: | Create storage / retrieval log | View storage / retrieval log | Modify storage / retrieval log | Cancel storage / retrieval log |
| Job owner | (note 1) | | | |
| U.ADMIN | | | | |
| U.NORMAL | | | denied | denied |
| Unauthenticated | (condition 1) | | denied | denied |

***Application Note:***

*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.*

*In particular, referring to Table 4 and Table 5:*

- *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.*

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

*See also the following Notes that are referenced in Table 4 and Table 5:*

***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.*

***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.*

***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.*

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

# 5.5. Identification and Authentication (FIA)

## 5.5.1. FIA_ATD.1 User attribute definition

(for O.USER_AUTHORIZATION)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

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

***Application Note:***

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

## 5.5.2. FIA_PMG_EXT.1 Extended: Password Management

(for O.USER_I&A)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

**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;

***Application Note:***

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

### 5.5.3. FIA_UAU.1 Timing of authentication

(for O.USER_I&A)
**Hierarchical to:**
No other components.

**Dependencies:**
FIA_UID.1 Timing of identification

**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.

**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.

***Application Note:***

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

### 5.5.4. FIA_UAU.7 Protected authentication feedback

(for O.USER_I&A)
**Hierarchical to:**
No other components.

**Dependencies:**
FIA_UAU.1 Timing of authentication

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

***Application Note:***

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

### 5.5.5. FIA_UID.1 Timing of identification

(for O.USER_I&A and O.ADMIN_ROLES)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

**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.

**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.

***Application Note:***

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

### 5.5.6. FIA_USB.1 User-subject binding

(for O.USER_I&A)
**Hierarchical to:**
No other components.

**Dependencies:**
FIA_ATD.1 User attribute definition

**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*].

**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*].

**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*].

## 5.6. Security Management (FMT)

### 5.6.1. 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

**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**.

### 5.6.2. 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

**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 authorized identified roles*].

### 5.6.3. 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

**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.

**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.

***Application Note:***

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

### 5.6.4. 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

**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 6.**

*Table 6. Management of TSF Data*

| Data | Operation | Authorized 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 |

### 5.6.5. 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.

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

***Application Note:***

Regarding “management functions provided by the TSF”, the ST Author should consider management functions that support the security objectives of this protection profile.

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

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

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

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

For O.UPDATE_VERIFICATION:

- Configuring firmware/software updates

For O.COMMS_PROTECTION:

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

For O.AUDIT:

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

For O.STORAGE_ENCRYPTION, O.KEY_MATERIAL:

- Configuring and invoking encryption of Nonvolatile Storage Devices

(Optional) For O.IMAGE_OVERWRITE, O.PURGE DATA:

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

### 5.6.6. 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

**FMT_SMR.1.1** The TSF shall maintain the roles **U.ADMIN, U.NORMAL**.

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

# 5.7. Privacy (FPR)

There are no class FPR requirements.

# 5.8. Protection of the TSF (FPT)

### 5.8.1. FPT_SBT_EXT.1 Extended: Secure Boot

(for O.FW_INTEGRITY)
**Hierarchical to:**
No other components.

**Dependencies:**
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/SigGen Cryptographic Operation (for signature generation/verification)
FCS_COP.1/KeyedHash Cryptographic Operation (for keyed-hash message authentication)
FCS_COP.1/DataEncryption Cryptographic Operation (Symmetric encryption/decryption)
FCS_COP.1/StorageEncryption Cryptographic operation (Data Encryption/Decryption)
FCS_COP.1/CMAC Cryptographic Operation (for keyed-hash message authentication)

**FPT_SBT_EXT.1.1** The TSF shall contain one or more chains of trust with each chain of trust anchored in a Root of Trust that is implemented in immutable code or a HW-based write-protection mechanism.

**FPT_SBT_EXT.1.2** At boot time the TSF shall use the chain(s) of trust to confirm integrity of its firmware/software using a [selection: *hash, digital signature, message authentication*] verification method.

**FPT_SBT_EXT.1.3** The TSF shall [selection: *enter maintenance mode, halt boot process, reboot the device, [assignment: another behavior of TOE]*] in the event of a boot time verification failure so that the corrupted firmware/software isn't executed.

**FPT_SBT_EXT.1.4** Following failure of verification, the TSF shall provide a mechanism to: [selection: *revert to previous TOE image, reinstall TOE image, perform a factory reset, indicate a need to contact vendor support*].

**FPT_SBT_EXT.1.5** The TSF shall contain [selection: *hash data, digital signature data, message authentication code, public key for digital signature, symmetric key for message authentication with confidentiality protection as defined in FPT_SBT_EXT.1.6*] in the Hardware Root of Trust.

**FPT_SBT_EXT.1.6** The TSF shall make the symmetric key accessible only to the Hardware Root of Trust

***Application Note:***

*The 'contact vendor support' may be selected to allow for the vendor to diagnose a failure in the verification carried out by the TOE at start-up. This selection provides a user of the TOE the option to involve the vendor in the resolution of a verification failure if performing the actions specified by any other selection does not change the state of the TOE.*

### 5.8.2. FPT_SKP_EXT.1 Extended: Protection of TSF Data

(for O.COMMS_PROTECTION)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

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

***Application Note:***

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.

### 5.8.3. FPT_STM.1 Reliable time stamps

(for O.AUDIT)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

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

***Application Note:***

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

### 5.8.4. FPT_TST_EXT.1 Extended: TSF testing

(for O.TSF_SELF_TEST)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

**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.

***Application Note:***

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/SigGen, or by hash specified in FCS_COP.1/Hash.

Self-test is intended to detect malfunctions which may compromise the TSF. Since the integrity of the firmware/software is guaranteed by FPT_SBT_EXT, the function for FPT_TST_EXT should address the malfunction detection like DRBG self-test defined in ISO/IEC 18031:2011.

### 5.8.5. FPT_TUD_EXT.1 Extended: Trusted Update

(for O.UPDATE_VERIFICATION)
**Hierarchical to:**
No other components.

**Dependencies:**
FCS_COP.1/SigGen Cryptographic Operation (for signature generation/verification),
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm).

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

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

**FPT_TUD_EXT.1.3** The TSF shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [selection: *published hash, no other functions*] prior to installing those updates.

***Application Note:***

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.

The digital signature mechanism is specified in FCS_COP.1/SigGen. The published hash is generated by one of the functions specified in FCS_COP.1/Hash. It is acceptable to implement both mechanisms.

# 5.9. Resource Utilization (FRU)

There are no class FRU requirements.

# 5.10. TOE Access (FTA)

## 5.10.1. FTA_SSL.3 TSF-initiated termination

(for O.USER_I&A)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

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

# 5.11. Trusted Paths/Channels (FTP)

## 5.11.1. 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_TLSC_EXT Extended: TLS Client Protocol and/or FCS_TLSS_EXT Extended: TLS Server Protocol, or
FCS_SSHC_EXT Extended: SSH Client Protocol or FCS_SSHS_EXT Extended: SSH Server Protocol, or
FCS_DTLSC_EXT Extended: DTLS Client Protocol and/or FCS_DTLSS_EXT Extended: DTLS Server Protocol, or
FCS_HTTPS_EXT.1 Extended: HTTPS selected ].

**FTP_ITC.1.1 Refinement:** The TSF shall **use [selection: IPsec, SSH, TLS, DTLS, 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**.

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

**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***].

***Application Note:***

*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.*

*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 Section A.2, "Protected Communications" 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.*

*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.*

*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.*

### 5.11.2. FTP_TRP.1/Admin Trusted path (for Administrators)

(for O.COMMS_PROTECTION)
**Hierarchical to:**
No other components.

**Dependencies:**
[ FCS_IPSEC_EXT.1 Extended: IPsec selected, or
FCS_TLSC Extended: TLS Client Protocol and/or FCS_TLSS_EXT Extended: TLS Server Protocol, or
FCS_SSHC_EXT Extended: SSH Client Protocol or FCS_SSHS_EXT Extended: SSH Server Protocol, or
FCS_DTLSC_EXT Extended: DTLS Client Protocol and/or FCS_DTLSS_EXT Extended: Server DTLS Protocol, or
FCS_HTTPS_EXT.1 Extended: HTTPS selected ].

**FTP_TRP.1.1/Admin Refinement:** The TSF shall **use [selection, choose at least one of: IPsec, SSH, TLS, DTLS, 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**.

**FTP_TRP.1.2/Admin Refinement:** The TSF shall permit **remote administrators** to initiate communication via the trusted path

**FTP_TRP.1.3/Admin Refinement:** The TSF shall require the use of the trusted path for **initial administrator authentication and all remote administration actions**.

***Application Note:***

*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 Section A.2, "Protected Communications" corresponding to their selection are copied to the ST if not already present.*

# 5.12. TOE Security Functional Requirements Rationale

*Table 7. TOE Security Functional Requirements Rationale*

| Objective | Addressed by | Rationale |
|---|---|---|
| **O.USER_AUTHORIZATION** | FDP_ACC.1 | This requirement defines an access control policy that governs the authorization required to interact with user data. |
| | FDP_ACF.1 | This requirement defines the rules enforced by the access control policy defined in FDP_ACC.1 to control access to user data. |

| Objective | Addressed by | Rationale |
|---|---|---|
| | FIA_ATD.1 | This requirement defines the list of security attributes belonging to individual users that supports user authentication. |
| | FMT_MSA.1 | This requirement enforces restrictions on the subjects that can interact with user data and their security attributes. |
| | FMT_MSA.3 | This requirement defines the default access restrictions that are enforced on user data security attributes if not overridden by specific access control policy rules. |
| | FMT_SMF.1 | This requirement defines the management functions that are provided by the TOE to authorized users. |
| | FMT_SMR.1 | This requirement defines the different security-related roles that the TOE recognizes. |
| **O.USER_I&A** | FIA_AFL.1 | This requirement defines how many consecutive unsuccessful authentication failures to prove a user's identity trigger actions by the TOE and what those actions will be. |
| | FIA_PMG_EXT.1 | This requirement defines the rules for passwords used by users for purposes of proving their identity to the TOE at the TOE itself. |
| | FIA_UAU.1 | This requirement defines the allowed actions that can be performed on behalf of a user before the user is authenticated and requires users to be authenticated before security functions by the TOE can be performed. |
| | FIA_UAU.7 | This requirement defines what type of feedback to the user is to be provided while authentication is in progress.. |
| | FIA_UID.1 | This requirement defines what actions users can perform before being identified by the TOE. |
| | FIA_USB.1 | This requirement defines the rules governing the association of the user's security attributes to a subject acting on the user's behalf. |
| | FTA_SSL.3 | This requirement enforces that the TOE terminates an interactive user session after a defined period of inactivity. |
| **O.ACCESS_CONTROL** | FDP_ACC.1 | This requirement defines an access control policy that governs the authorization required to interact with user data. |
| | FDP_ACF.1 | This requirement defines the rules enforced by the access control policy defined in FDP_ACC.1 to control access to user data. |
| | FMT_MSA.1 | This requirement enforces restrictions on the subjects that can interact with user data and their security attributes. |
| | FMT_MSA.3 | This requirement defines the default access restrictions that are enforced on user data security attributes if not overridden by specific access control policy rules. |
| | FMT_MTD.1 | This requirement defines the roles that can perform specified operations on TSF data. |

| Objective | Addressed by | Rationale |
|---|---|---|
| | FMT_SMF.1 | This requirement defines the management functions that are provided by the TOE to authorized users. |
| | FMT_SMR.1 | This requirement defines the different security-related roles that the TOE recognizes. |
| **O.ADMIN_ROLES** | FIA_UID.1 | This requirement defines what admin actions users can perform before being identified by the TSF. |
| | FMT_MOF.1 | This requirement enforces access control on the admin functions provided by the TOE. |
| | FMT_SMF.1 | This requirement defines the management functions that are provided by the TOE to authorized users. |
| | FMT_SMR.1 | This requirement defines the different security-related roles that the TOE recognizes. |
| | FPT_STM.1 | This requirement provides reliable system time services that are used to provide time stamps on audit log records. |
| **O.UPDATE_VERIFICATION** | FCS_COP.1/SigGen | This requirement defines the cryptographic algorithms that must be applied to generate digital signatures that are used verify the integrity of software/firmware upgrade files for the TOE. |
| | FCS_COP.1/Hash | This requirement defines the cryptographic algorithms that must be applied to generate cryptographic hash values that are used to verify the integrity of software/firmware upgrade files for the TOE. |
| | FPT_TUD_EXT.1 | This requirement defines the ability of the admin to initiate and verify firmware/software updates to the TOE. |
| **O.TSF_SELF_TEST** | FPT_TST_EXT.1 | This requirement enforces the use of self-tests during initial start-up (and power on) to demonstrate the correct operation of the TOE. |
| **O.COMMS_PROTECTION** | FCS_CKM.1/SKG | This requirement generates the symmetric keys needed to encrypt data being transmitted to/from the TOE. |
| | FCS_CKM.2 | This requirement provides the methods for performing key establishment between the TOE and IT entity that data is to be transferred either to the TOE or from the TOE. |
| | FCS_CKM_EXT.4 | This requirement enforces that all plaintext secret and private cryptographic keys and cryptographic critical security parameters must be destroyed when no longer needed. |
| | FCS_CKM.4 | This requirement enforces the methods that must be used to destroy all cryptographic keys. |
| | FCS_COP.1/DataEncryption | This requirement defines the cryptographic algorithms that must be applied to encrypt/decrypt data that is to be transmitted to/from the TOE. |

| Objective | Addressed by | Rationale |
|---|---|---|
| | FCS_COP.1/SigGen | This requirement defines the cryptographic algorithms that must be applied to generate digital signatures that help to verify the integrity of data transmitted to/from the TOE. |
| | FCS_RBG_EXT.1 | This requirement defines the random bit generation mechanisms that must be applied to generate cryptographic keys and cryptographic critical security parameters. |
| | FPT_SKP_EXT.1 | This requirement enforces the prevention of reading all pre-shared keys, symmetric keys, and private keys. |
| | FTP_ITC.1 | This requirement provides for a trusted communications channel to transmit the user and TSF data between the TOE and a trusted external IT entity. |
| | FTP_TRP.1/Admin | This requirement provides for a trusted communications path between the TOE and the admin |
| | FCS_IPSEC_EXT.1 | This requirement defines the IPsec protocol for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FCS_TLSC_EXT.1 | This requirement defines the TLS protocol acting as a client without mutual authentication for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FCS_TLSS_EXT.1 | This requirement defines the TLS protocol acting as a server without mutual authentication for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FCS_SSHC_EXT.1 | This requirement defines the SSH protocol acting as a client for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FCS_SSHS_EXT.1 | This requirement defines the SSH protocol acting as a server for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FCS_HTTPS_EXT.1 Extended | This requirement defines the TLS/HTTPS protocol for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FCS_COP.1/KeyedHash | This requirement defines the cryptographic algorithms that must be applied to perform keyed-hash message authentication. |
| | FIA_PSK_EXT.1 Extended | This requirement defines the components of pre-shared keys for the IPsec protocol. |
| | FCS_DTLSC_EXT.1 | This requirement defines the DTLS protocol acting as a client without mutual authentication for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |

| Objective | Addressed by | Rationale |
|---|---|---|
| | FCS_DTLSS_EXT.1 | This requirement defines the DTLS protocol acting as a server without mutual authentication for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FIA_X509_EXT.1 | This requirement defines the rules for the validation of X.509 certificates. |
| | FIA_X509_EXT.2 | This requirement defines the use of X.509 certificates for authentication of the protocols used for secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FIA_X509_EXT.3 | This requirement defines the rules for a certificate request for an X.509 certificate. |
| | FCS_CKM.1/AKG | This requirement defines the cryptographic algorithms that must be applied to generate asymmetric cryptographic keys. |
| | FTP_TRP.1/NonAdmin | This requirement provides for a trusted communications path between the TOE and a user who is not an admin. |
| | FCS_DTLSC_EXT.2 | This requirement defines the DTLS protocol acting as a client with mutual authentication for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FCS_DTLSS_EXT.2 | This requirement defines the DTLS protocol acting as a server with mutual authentication for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FCS_TLSC_EXT.2 | This requirement defines the TLS protocol acting as a client with mutual authentication for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| | FCS_TLSS_EXT.2 | This requirement defines the TLS protocol acting as a server with mutual authentication for the secure transmission of user and TSF data between the TOE and a trusted external IT entity. |
| **O.AUDIT** | FAU_GEN.1 | This requirement defines the minimum required auditable events and the required contents of each audit record. |
| | FAU_GEN.2 | This requirement enforces associating each auditable event with the identity of the user that caused the event |
| | FAU_SAR.1 | This requirement provides for the reading of audit records in an interpretable manner. |
| | FAU_SAR.2 | This requirement enforces only allowing users with explicit read-access to read audit records. |
| | FAU_STG.1 | This requirement enforces protection of stored audit records from unauthorized deletion or modification. |
| | FAU_STG.4 | This requirement defines actions to be taken when the audit log is full. |

| Objective | Addressed by | Rationale |
|---|---|---|
| | FAU_STG_EXT.1 | This requirement provides for the transmission of audit log records to a trusted external IT entity over a trusted communications channel. |
| | FTP_ITC.1 | This requirement provides for a trusted communications channel to transmit the audit log records between the TOE and a trusted external IT entity. |
| **O.STORAGE_ENCRYPTION** | FCS_CKM.1/SKG | This requirement generates the symmetric keys needed to encrypt data being stored on the TOE. |
| | FCS_CKM_EXT.4 | This requirement enforces that all plaintext secret and private cryptographic keys and cryptographic critical security parameters must be destroyed when no longer needed. |
| | FCS_CKM.4 | This requirement enforces the methods that must be used to destroy all cryptographic keys. |
| | FCS_COP.1/Hash | This requirement defines the cryptographic algorithms that must be applied to generate cryptographic hash values that are used verify the integrity of user and TSF data stored on the TOE. |
| | FCS_COP.1/DataEncryption | This requirement defines the cryptographic algorithms that must be applied to encrypt/decrypt user and TSF data. |
| | FCS_RBG_EXT.1 | This requirement defines the random bit generation mechanisms that must be applied to generate cryptographic keys and cryptographic critical security parameters used to protect the confidentiality and integrity of user and TSF data stored on the TOE. |
| | FCS_COP.1/StorageEncryption | This requirement defines the cryptographic algorithms that must be applied to encrypt/decrypt user and TSF data stored on the TOE. |
| | FCS_COP.1/KeyWrap | This requirement defines the cryptographic algorithms that must be applied to perform key wrapping of a cryptographic key in support of key chaining. |
| | FCS_COP.1/KeyEnc | This requirement defines the cryptographic algorithms that must be applied to perform key encryption and decryption. |
| | FCS_COP.1/KeyTransport | This requirement defines the cryptographic algorithms that must be applied to transfer a key and other keying material from an entity that either owns the key or generates the key to another entity that is intended to use the key. |
| | FCS_SMC_EXT.1 | This requirement defines the methods for combining submasks to generate an intermediary key. |
| | FCS_KDF_EXT.1 Extended | This requirement defines the methods for generating an intermediary key. |

| Objective | Addressed by | Rationale |
|---|---|---|
| | FCS_PCC_EXT.1 Extended | This requirement defines the cryptographic algorithms that must be applied to construct cryptographic passwords. |
| | FCS_COP.1/CMAC | This requirement defines the cryptographic algorithms that must be applied to perform message authentication. |
| | FCS_SNI_EXT.1 Extended | This requirement defines the creation and usage of salts, nonces and IVs. |
| | FCS_KYC_EXT.1 Extended | This requirement defines the rules for creating a key chain to unlock a self-encrypting drive. |
| | FDP_DSK_EXT.1 Extended | This requirement defines the rules for the protection of user and TSF data stored on the TOE. |
| **O.KEY_MATERIAL** | FPT_KYP_EXT.1 Extended | This requirement defines the rules for the protection of cryptographic keys and key material. |
| **O.FAX_NET_SEPARATION** (conditionally mandatory) | FDP_FXS_EXT.1 Extended | This requirement enforces the prohibition of communication via the fax interface, except for transmitting or receiving User Data using fax protocols. |
| **O.IMAGE_OVERWRITE** (optional) | FDP_RIP.1/Overwrite | This requirement enforces the overwriting of user document data stored on the TOE after each job is processed or cancelled. |
| **O.WIPE_DATA** (optional) | FCS_CKM_EXT.4 | This requirement enforces that all plaintext secret and private cryptographic keys and cryptographic critical security parameters must be destroyed when no longer needed. |
| | FCS_CKM.4 | This requirement enforces the methods that must be used to destroy all cryptographic keys. |
| | FDP_WIPE_EXT.1 | This requirement enforces that customer-supplied user and TSF data is made unavailable at the request of the admin. |
| **O.AUTH_FAILURES** | FIA_AFL.1 | This requirement defines how many consecutive unsuccessful authentication failures to prove a user's identity trigger actions by the TOE and what those actions will be. |
| **O.FW_INTEGRITY** | FPT_SBT_EXT.1 | This requirement defines how the integrity of firmware/software at boot time is to be verified via chains of trust, each one anchored in its own root of trust. |
| **O.STRONG_CRYPTO** | FCS_CKM.1/SKG | This requirement ensures the generation of strong symmetric keys. |
| | FCS_CKM.2 | This requirement ensures the use of strong key establishment mechanisms. |

| Objective | Addressed by | Rationale |
|---|---|---|
| | FCS_COP.1/DataEncryption | This requirement ensures the use of strong methods to perform data encryption/decryption. |
| | FCS_COP.1/SigGen | This requirement ensures the use of strong digital signature services. |
| | FCS_COP.1/Hash | This requirement ensures the use of strong hash mechanisms. |
| | FCS_RBG_EXT.1 | This requirement ensures the use of strong random bit generation mechanisms. |
| | FPT_STM.1 | This requirement provides reliable system time services that may be used as inputs to cryptographic functions. |
| | FCS_COP.1/StorageEncryption | This requirement ensures the use of strong methods to perform data encryption/decryption. |
| | FCS_COP.1/KeyWrap | This requirement ensures the use of strong methods to perform key wrapping. |
| | FCS_COP.1/KeyEnc | This requirement ensures the use of strong methods to perform key encryption. |
| | FCS_COP.1/KeyTransport | This requirement ensures the use of strong methods to perform key transport. |
| | FCS_SMC_EXT.1 | This requirement ensures the use of strong methods to perform submask combining. |
| | FCS_IPSEC_EXT.1 | This requirement defines the implementation of IPsec using strong cryptography. |
| | FCS_TLSC_EXT.1 | This requirement defines the implementation of TLS acting as a client without mutual authentication using strong cryptography. |
| | FCS_TLSS_EXT.1 | This requirement defines the implementation of TLS acting as a server without mutual authentication using strong cryptography. |
| | FCS_SSHC_EXT.1 | This requirement defines the implementation of SSH acting as a client without mutual authentication using strong cryptography. |
| | FCS_SSHS_EXT.1 | This requirement defines the implementation of SSH acting as a server without mutual authentication using strong cryptography. |
| | FCS_HTTPS_EXT.1 Extended | This requirement defines the implementation of TLS/HTTPS using strong cryptography. |
| | FCS_COP.1/KyedHash | This requirement ensures the use of strong methods to perform keyed-hash message authentication. |

| Objective | Addressed by | Rationale |
|---|---|---|
| | FCS_DTLSC_EXT.1 | This requirement defines the implementation of DTLS acting as a client without mutual authentication using strong cryptography. |
| | FCS_DTLSS_EXT.1 | This requirement defines the implementation of DTLS acting as a server without mutual authentication using strong cryptography. |
| | FCS_PCC_EXT.1 Extended | This requirement ensures the use of strong methods to generate cryptographic passwords. |
| | FCS_KDF_EXT.1 Extended | This requirement ensures the use of strong methods for performing cryptographic key derivation. |
| | FCS_COP.1/CMAC | This requirement ensures the use of strong methods to perform message authentication. |
| | FCS_SNI_EXT.1 Extended | This requirement ensures that salts and nonces used by the TOE do not negatively impact key strength. |
| | FCS_CKM.1/AKG | This requirement ensures the generation of strong asymmetric keys. |
| | FPT_KYP_EXT.1 Extended | This requirement ensures the use of strong methods to protection of key and key material. |
| | FCS_KYC_EXT.1 Extended | This requirement ensures the use of strong methods to perform key chaining. |
| | FCS_DTLSC_EXT.2 | This requirement defines the implementation of DTLS acting as a client with mutual authentication using strong cryptography. |
| | FCS_DTLSS_EXT.2 | This requirement defines the implementation of DTLS acting as a server with mutual authentication using strong cryptography. |
| | FCS_TLSC_EXT.2 | This requirement defines the implementation of TLS acting as a client with mutual authentication using strong cryptography. |
| | FCS_TLSS_EXT.2 | This requirement defines the implementation of TLS acting as a server with mutual authentication using strong cryptography. |

# 6. Security Assurance Requirements

The Section 4, “Security Objectives” for the TOE were constructed to address [threats] identified in the Section 3, “Security Problem Definition”. The Section 5, “Security Functional Requirements” are a formal instantiation of the Section 4, “Security Objectives”. This cPP identifies the Security Assurance Requirements to frame the extent to which the evaluator assesses the documentation applicable for the evaluation and performs independent testing.

This section lists the set of SARs from CC part 3 that are required in evaluations against this cPP. Individual Evaluation Activities to be performed are specified in [SD].

The general model for evaluation of TOEs against STs written to conform to this cPP is as follows:

After the ST has been approved for evaluation, the ITSEF (IT Security Evaluation Facility) will obtain the TOE, supporting environmental IT (if required), and the administrative/user guides for the TOE. The ITSEF is expected to perform actions mandated by the Common Evaluation Methodology (CEM) for the ASE and ALC SARs. The ITSEF also performs the Evaluation Activities contained within the SD, which are intended to be an interpretation of the other CEM assurance requirements as they apply to the specific technology instantiated in the TOE. The Evaluation Activities that are captured in the SD also provide clarification as to what the developer needs to provide to demonstrate the TOE is compliant with the cPP.

*Table 8. Security Assurance Requirements*

| Assurance Class | Assurance Components |
|---|---|
| Security Target (ASE) | Conformance Claims (ASE_CCL.1) |
| | Extended components definition (ASE_ECD.1) |
| | ST introduction (ASE_INT.1) |
| | Security objectives for the operational environment (ASE_OBJ.1) |
| | Stated security requirements (ASE_REQ.1) |
| | Security Problem Definition (ASE_SPD.1) |
| | TOE summary specification (ASE_TSS.1) |
| Development (ADV) | Basic functional specification (ADV_FSP.1) |
| Guidance documents (AGD) | Operational user guidance (AGD_OPE.1) |
| | Preparative procedures (AGD_PRE.1) |
| Life cycle support (ALC) | Labeling of the TOE (ALC_CMC.1) |
| | TOE CM coverage (ALC_CMS.1) |
| Tests (ATE) | Independent testing – conformance (ATE_IND.1) |
| Vulnerability assessment (AVA) | Vulnerability survey (AVA_VAN.1) |

## 6.1. ASE: Security Target

The ST is evaluated as per ASE activities defined in the [CEM]. In addition, there may be Evaluation Activities specified within the [SD] that call for necessary descriptions to be included in the TSS that are specific to the TOE technology type.

Appendix E, *Entropy Documentation and Assessment* provides a description of the information expected to be provided regarding the quality of entropy in the random bit generator.

Given the criticality of the key management scheme, this cPP 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, *Key Management Document* for details on the expectation of the developer's Key Management Description.

## 6.2. ADV: Development

The design information about the TOE is contained in the guidance documentation available to the end user as well as the TSS portion of the ST, and any additional information required by this cPP that is not to be made public (e.g., Entropy Report).

### 6.2.1. Basic Functional Specification (ADV_FSP.1)

The functional specification describes the TOE Security Functions Interfaces (TSFIs). It is not necessary to have a formal or complete specification of these interfaces. Additionally, because TOEs conforming to this cPP will necessarily have interfaces to the Operational Environment that are not directly invokable by TOE users, there is little point specifying that such interfaces be described in and of themselves since only indirect testing of such interfaces may be possible. For this cPP, the Evaluation Activities for this family 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" documentation is necessary to satisfy the Evaluation Activities specified in [SD].

The Evaluation Activities in [SD] are associated with the applicable SFRs; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

## 6.3. AGD: Guidance Documentation

The guidance documents will be provided with the ST. Guidance must include a description of how the IT personnel verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by the IT personnel.

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 TSF in that environment; and
- instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and
- instructions to provide a protected administrative capability.

Guidance pertaining to particular security functionality must also be provided; requirements on such guidance are contained in the Evaluation Activities specified in the [SD].

### 6.3.1. Operational User Guidance (AGD_OPE.1)

The operational user guidance does not have to be contained in a single document. Guidance to users, administrators and application developers can be spread among documents or web pages.

The developer should review the Evaluation Activities contained in the [SD] to ascertain the specifics of the guidance that the evaluator will be checking for. This will provide the necessary information for the preparation of acceptable guidance.

### 6.3.2. Preparative Procedures (AGD_PRE.1)

As with the operational guidance, the developer should look to the Evaluation Activities to determine the required content with respect to preparative procedures.

## 6.4. Class ALC: Life-cycle Support

At the assurance level provided for TOEs conformant to this cPP, 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 is a reflection on the information to be made available for evaluation at this assurance level.

### 6.4.1. Labelling of the TOE (ALC_CMC.1)

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

### 6.4.2. TOE CM Coverage (ALC_CMS.1)

Given the scope of the TOE and its associated evaluation evidence requirements, the evaluator performs the CEM work units associated with ALC_CMC.1.

## 6.5. Class ATE: Tests

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

### 6.5.1. Independent Testing – Conformance (ATE_IND.1)

Testing is performed to confirm the functionality described in the TSS as well as the operational guidance (includes "evaluated configuration" instructions). The focus of the testing is to confirm that the requirements specified in Section 5 are being met. The Evaluation Activities in the SD identify the specific testing activities necessary to verify compliance with the SFRs. The evaluator produces a test report documenting the plan for and results of testing, as well as coverage arguments focused on the platform/TOE combinations that are claiming conformance to this cPP.

## 6.6. Class AVA: Vulnerability Assessment

For the first generation of this cPP, the iTC is expected to survey open sources to discover what vulnerabilities have been discovered in these types of products and provide that content into the AVA_VAN discussion. In most cases, these vulnerabilities will require sophistication beyond that of a basic attacker. This information will be used in the development of future protection profiles.

### 6.6.1. Vulnerability Survey (AVA_VAN.1)

[SD] provides a guide to the evaluator in performing a vulnerability analysis.

# Appendix A: Selection-Based Requirements

## A.1. Confidential Data on Nonvolatile Storage Devices

### A.1.1. FCS_COP.1/StorageEncryption Cryptographic operation (Data Encryption/Decryption)

(for O.STORAGE_ENCRYPTION, O.STRONG_CRYPTO)
**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/SKG Cryptographic Key Generation (Symmetric Keys)]
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**FCS_COP.1.1/StorageEncryption** The TSF shall perform **data encryption and decryption** in accordance with a specified cryptographic algorithm [selection:

- *AES used in* [selection: *CBC, GCM, XTS*] *mode*,
- SEED used in [selection:
  - CBC mode with non-repeating and unpredictable IVs,
  - CFB mode with non-repeating and unpredictable IVs,
  - OFB mode with unique IVs,
  - CTR mode with unique, incremental counter,
  - CCM mode with unpredictable, non-repeating nonce, minimum size of 64 bits,
  - GCM mode with non-repeating IVs],
- HIGHT used in [selection:
  - CBC mode with non-repeating and unpredictable IVs,
  - CFB mode with non-repeating and unpredictable IVs,
  - OFB mode with unique IVs,
  - CTR mode with unique, incremental counter],
- LEA used in [selection:
  - CBC mode with non-repeating and unpredictable IVs,
  - CFB mode with non-repeating and unpredictable IVs,

    - OFB mode with unique IVs,
    - CTR mode with unique, incremental counter,
    - CCM mode with unpredictable, non-repeating nonce, minimum size of 64 bits,
    - GCM mode with non-repeating IVs]

] and cryptographic key sizes [selection:

Case: AES algorithm

- [selection: 128 bits, 192 bits, 256 bits],

Case: SEED algorithm

- 128 bits,

Case: HIGHT algorithm

- 128 bits,

Case: LEA algorithm

- [selection: 128 bits, 192 bits, 256 bits]

] that meet the following [selection:

Case: AES algorithm

- *ISO 18033-3*, [selection: *CBC as specified in ISO 10116, GCM as specified in ISO/IEC 19772, and XTS as specified in IEEE 1619*],

Case: SEED algorithm

- SEED as specified in ISO/IEC 18033-3:2010, [selection: CBC as specified in ISO/IEC 10116:2017 (clause 7), CFB as specified in ISO/IEC 10116:2017 (clause 8), OFB as specified in ISO/IEC 10116:2017 (clause 9), CTR as specified in ISO/IEC 10116:2017 (clause 10), CCM as specified in ISO/IEC 19772:2009 (clause 8), GCM as specified in ISO/IEC 19772:2009 (clause 11) and NIST SP800-38D],

Case: HIGHT algorithm

- HIGHT as specified in ISO/IEC 18033-3:2010, [selection: CBC as specified in ISO/IEC 10116:2017 (clause 7), CFB as specified in ISO/IEC 10116:2017 (clause 8), OFB as specified in ISO/IEC 10116:2017 (clause 9), CTR as specified in ISO/IEC 10116:2017 (clause 10)],

Case: LEA algorithm

- LEA as specified in ISO/IEC 29192-2:2019, [selection: CBC as specified in ISO/IEC 10116:2017 (clause 7), CFB as specified in ISO/IEC 10116:2017 (clause 8), OFB as specified in ISO/IEC 10116:2017 (clause 9), CTR as specified in ISO/IEC 10116:2017 (clause 10), CCM as specified in ISO/IEC 19772:2009 (clause 8), GCM as specified in ISO/IEC 19772:2009 (clause 11) and NIST SP800-38D]

]

***Application Note:***

*This cPP allows for firmware/software encryption or hardware encryption.*

*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.*

*If either SEED in GCM mode with non-repeating IVs or LEA in GCM mode with non-repeating IVs is selected, IV length must be equal to 96 bits, the deterministic IV construction method [SP800-38D, Section 8.2.1] must be used, and the MAC length t must be one of the values 96, 104, 112, 120, and 128 bits.*

*The intent of this requirement is to specify the approved modes that the ST Author may select for encryption of the appropriate information on the 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/SKG. 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.*

## A.1.2. FCS_COP.1/KeyWrap Cryptographic operation (Key Wrapping)

(selected in FCS_KYC_EXT.1.1, for O.STORAGE_ENCRYPTION, O.STRONG_CRYPTO)
**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/SKG Cryptographic Key Generation (Symmetric Keys) ]
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**FCS_COP.1.1/KeyWrap Refinement:** The TSF shall perform **key wrapping** in accordance with a specified cryptographic algorithm [selection: **AES in the following modes [selection: *KW, KWP, GCM, CCM*], SEED in the following modes [selection: Key Wrap, KWP, GCM, CCM], LEA in the following modes [selection: Key Wrap, KWP, GCM, CCM]]** and the cryptographic key size **[selection: *128 bits (AES, SEED, LEA), 192 bits (AES, LEA), 256 bits (AES, LEA)*]** that meet the following: [selection:

Case: AES algorithm

- [ISO/IEC 18033-3 (AES), [selection: *NIST SP 800-38F, ISO/IEC 19772, no other standards*]],

Case: SEED algorithm

- [ISO/IEC 18033-3:2010 (SEED), [selection: ISO/IEC 19772:2009, NIST SP 800-38F, sec. 6.3]],

Case: LEA algorithm

- [ISO/IEC 29192-2:2019 (LEA), [selection: ISO/IEC 19772:2009, NIST SP 800-38F, sec. 6.3]]

].

***Application Note:***

*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.*

### A.1.3. FCS_COP.1/KeyEnc Cryptographic operation (Key Encryption)

(selected from FCS_KYC_EXT.1.1, for O.STORAGE_ENCRYPTION, O.STRONG_CRYPTO)
**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/SKG Cryptographic Key Generation (Symmetric Keys)]
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**FCS_COP.1.1/KeyEnc Refinement:** The TSF shall perform **key encryption and decryption** in accordance with a specified cryptographic algorithm [selection:

Case: AES algorithm

- AES used in [[selection: *CBC*, *GCM*] mode] and cryptographic key sizes [selection: *128 bits, 192 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*],

Case: SEED algorithm

- SEED used in [selection: CCM, GCM] and cryptographic key size [selection: 128 bits, 256 bits] that meet the following: SEED as specified in ISO/IEC 18033-3:2010, [selection: CCM as specified in ISO/IEC 19772:2009, GCM as specified in ISO/IEC 19772:2009],

Case: LEA algorithm

- LEA used in [selection: CCM, GCM] and cryptographic key size [selection: 128 bits, 256 bits] that meet the following: LEA as specified in ISO/IEC 29192-2:2019, [selection: CCM as specified in ISO/IEC 19772:2009, GCM as specified in ISO/IEC 19772:2009]

].

***Application Note:***

*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.*

### A.1.4. FCS_COP.1/KeyTransport Cryptographic operation (Key Transport)

(selected in FCS_KYC_EXT.1.1 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/AKG Cryptographic Key Generation (Asymmetric Keys)]
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**FCS_COP.1.1/KeyTransport 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 bits, 3072 bits, 4096 bits, 6144 bits, 8192 bits*]** that meet the following: **NIST SP 800-56B, Revision 1**.

***Application Note:***

*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.*

### A.1.5. FCS_SMC_EXT.1 Extended: Submask Combining

(selected in FCS_KYC_EXT.1.1, for O.STORAGE_ENCRYPTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

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, BEV or DEK.

***Application Note:***

*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/Hash in* *Section 5.3.7, "FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)"*

## A.2. Protected Communications

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.

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.

## A.2.1. FCS_IPSEC_EXT.1 Extended: IPsec selected

(selected in FTP_ITC.1.1, FTP_TRP.1.1/Admin, FTP_TRP.1.1/NonAdmin, for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**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_CKM.1/AKG Cryptographic Key Generation (Asymmetric Keys),
FCS_COP.1/DataEncryption Cryptographic Operation (Symmetric encryption/decryption),
FCS_COP.1/SigGen Cryptographic Operation (for signature generation/verification),
FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm),
FCS_COP.1/KeyedHash Cryptographic Operation (for keyed-hash message authentication),
FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

***Application Note:***

*The TOE is required to use the IPsec protocol to establish connections used to communicate with an IPsec Peer.*

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

***Application Note:***

*RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a 'traditional' SPD, etc. Regardless of the implementation details, there is a notion of a 'rule' that a packet is 'matched' against and a resulting action that takes place.*

*While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the SPD can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface), but this is not required.*

**FCS_IPSEC_EXT.1.2** The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched and discards it.

**FCS_IPSEC_EXT.1.3** The TSF shall implement [selection: *transport mode, tunnel mode*].

***Application Note:***

*The ST author selects the supported modes of operation for IPsec.*

**FCS_IPSEC_EXT.1.4** The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [selection: AES-CBC-128 (RFC 3602), AES-CBC-192 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-192 (RFC 4106), AES-GCM-256 (RFC 4106)] together with a Secure Hash Algorithm (SHA)-based HMAC [selection: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, no HMAC algorithm].

***Application Note:***

*When an AES-CBC algorithm is selected, at least one SHA-based HMAC must also be chosen. If only an AES-GCM algorithm is selected, then a SHA-based HMAC is not required since AES-GCM satisfies both confidentiality and integrity functions. IPsec may utilise a truncated version of the SHA-based HMAC functions contained in the selections. Where a truncated output is utilised, it shall be highlighted in the TSS.*

**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 RFC 5996 and [selection: with no support for NAT traversal, with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]

].

***Application Note:***

*If the TOE implements SHA-2 hash algorithms for IKEv1 or IKEv2, the ST author selects RFC 4868. If the TOE implements the use of truncated SHA-based HMACs as described in RFC 4868, they shall be highlighted in the TSS.*

**FCS_IPSEC_EXT.1.6** The TSF shall ensure the encrypted payload in the [selection: IKEv1, IKEv2] protocol uses the cryptographic algorithms [selection: AES-CBC-128, AES-CBC-192, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES GCM-192, AES-GCM-256 (specified in RFC 5282)].

***Application Note:***

*AES-GCM-128, AES-GCM-192 and AES-GCM-256 may only be selected if IKEv2 is also selected, as there is no RFC defining AES-GCM for IKEv1.*

**FCS_IPSEC_EXT.1.7** The TSF shall ensure that [selection:

- IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on [selection:
  - number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range including 24] hours; ];

- IKEv2 SA lifetimes can be configured by a Security Administrator based on [selection:
  - number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range including 24] hours ]

].

***Application Note:***

*The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either volume-based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the guidance documentation generated for AGD_OPE.*

**FCS_IPSEC_EXT.1.8** The TSF shall ensure that [selection:

- IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [selection:
  - number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range including 8] hours; ];
- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [selection:
  - number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range including 8] hours; ]

].

***Application Note:***

*The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either volume-based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the guidance documentation generated for AGD_OPE.*

**FCS_IPSEC_EXT.1.9** The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [assignment: (one or more) number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group] bits.

***Application Note:***

*For DH groups 19 and 20, the 'x' value is the point multiplier for the generator point G.*

*Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.9 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ('bits of security') associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 is 192.*

**FCS_IPSEC_EXT.1.10** The TSF shall generate nonces used in [selection: IKEv1, IKEv2] exchanges of length [selection:

- according to the security strength associated with the negotiated Diffie-Hellman group;
- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash

].

***Application Note:***

*The ST author must select the second option for nonce lengths if IKEv2 is also selected (as this is mandated in RFC 5996). The ST author may select either option for IKEv1.*

*For the first option for nonce lengths, since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.10 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.*

*Because nonces may be exchanged before the DH group is negotiated, the nonce used should be large enough to support all TOE-chosen proposals in the exchange.*

**FCS_IPSEC_EXT.1.11** The TSF shall ensure that IKE protocols implement DH Group(s) [selection:

- [selection: 14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP), 17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526,
- [selection: 19 (256-bit Random ECP), 20 (384-bit Random ECP), 21 (521-bit Random ECP), 24 (2048-bit MODP with 256-bit POS)] according to RFC 5114.

].

***Application Note:***

*The selections are used to specify additional DH groups supported. This applies to IKEv1 and IKEv2 exchanges.*

**FCS_IPSEC_EXT.1.12** The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1

Phase 1, IKEv2 IKE_SA] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1 Phase 2, IKEv2 CHILD_SA] connection.

***Application Note:***

*The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. Obviously, the IKE version(s) chosen should be consistent not only in this element, but with other choices for other elements in this component. While it is acceptable for this capability to be configurable, the default configuration in the evaluated configuration (either 'out of the box' or by configuration guidance in the AGD documentation) must enable this functionality.*

**FCS_IPSEC_EXT.1.13** The TSF shall ensure that all IKE protocols perform peer authentication using [selection: RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [selection: Pre-shared Keys, no other method].

***Application Note:***

*At least one public-key-based Peer Authentication method is required in order to conform to this cPP; one or more of the public key schemes is chosen by the ST author to reflect what is implemented. The ST author also ensures that appropriate FCS requirements reflecting the algorithms used (and key generation capabilities, if provided) are listed to support those methods. Note that the TSS will elaborate on the way in which these algorithms are to be used (for example, RFC 2409 specifies three authentication methods using public keys; each one supported will be described in the TSS).*

**FCS_IPSEC_EXT.1.14** The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: [selection: SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN, CN: IP address, CN: Fully Qualified Domain Name (FQDN), CN: user FQDN, Distinguished Name (DN)] and [selection: no other reference identifier type, [assignment: other supported reference identifier types]].

***Application Note:***

*When using RSA or ECDSA certificates for peer authentication, the reference and presented identifiers take the form of either a DN, IP address, FQDN or user FQDN. The reference identifier is the identifier the TOE expects to receive from the peer during IKE authentication. The presented identifier is the identifier that is contained within the peer certificate body. The ST author shall select the presented and reference identifier types supported and may optionally assign additional supported identifier types in the second selection. Excluding the DN identifier type (which is necessarily the Subject DN in the peer certificate), the TOE may support the identifier in either the Common Name or Subject Alternative Name (SAN) or both.*

*The critical requirement of X.509 identifiers is the ability to bind the public key uniquely to an identity. This can be achieved by using strongly-typed identifiers or controlling the CA and certificate issuance. One recommended method for identity verification is supporting the use of the Subject Alternative Name (SAN) extension using DNS names, URI names, or Service Names. However, the support for a SAN extension is optional as long as identifier uniqueness can be achieved by other means.*

*In a future version of this cPP, SAN and/or DN support might be required for all TOEs, support for CN*

*might be optional, and the "other supported referenced identifier types" selection might be removed. In a future version of this cPP, it might also be required that the SAN (when present) shall take precedence over CN.*

*Supported peer certificate algorithms are the same as FCS_IPSEC_EXT.1.13*

### A.2.2. FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol

TLS is not a required component of this cPP. If a TOE implements TLS, a corresponding selection in FTP_ITC.1, FTP_TRP.1/Admin, and/or FTP_TRP.1/NonAdmin should be made to define what the TLS protocol is implemented to protect. If a corresponding option to support TLS has been selected in at least one of the SFRs named above, the corresponding selection-based TLS-related SFRs should be added to the ST from Section A.2.2, "FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol" (i.e. FCS_TLSC_EXT.1 and/or FCS_TLSS_EXT.1).

The support for mutual authentication is optional when using TLS. If a TOE implements TLS with mutual authentication the corresponding optional SFRs should be added to the ST from chap. Section C.3.2, "FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol" (i.e. FCS_TLSC_EXT.2 and/or FCS_TLSS_EXT.2) in addition to the corresponding SFRs from Section A.2.2, "FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol".

A TOE may act as the client, the server, or both in TLS sessions. The requirement has been separated into TLS Client (FCS_TLSC_EXT) and TLS Server (FCS_TLSS_EXT) requirements to allow for these differences. If the TOE acts as the client during the claimed TLS sessions, the ST author should claim the corresponding FCS_TLSC_EXT requirements. If the TOE acts as the server during the claimed TLS sessions, the ST author should claim the corresponding FCS_TLSS_EXT requirements. If the TOE acts as both a client and server during the claimed TLS sessions, the ST author should claim the corresponding FCS_TLSC_EXT and FCS_TLSS_EXT requirements.

Additionally, TLS may or may not be performed with client authentication. The ST author shall claim FCS_TLSC_EXT.1 and/or FCS_TLSS_EXT.1 if the TOE does not support client authentication. The ST author should claim FCS_TLSC_EXT.2 and/or FCS_TLSS_EXT.2 if client authentication is performed by the TOE.

The following list contains all DTLS-/TLS-related ciphersuites supported by this cPP.

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246

- TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

List 1: List of supported TLS-related ciphersuites

#### A.2.2.1. FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication

(for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_TLSC_EXT.1.1** The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [selection: select supported ciphersuites from List 1] and no other ciphersuites.

***Application Note:***

*The ciphersuites to be tested in the evaluated configuration are limited by this requirement and must be selected from the ciphersuites defined in List 1. The ST author should select the ciphersuites that are supported. Even though RFC 5246 mandates implementation of specific ciphers, there is no*

*requirement to implement TLS_RSA_WITH_AES_128_CBC_SHA in order to claim conformance to this cPP.*

*These requirements will be revisited as new TLS versions are standardized by the IETF.*

*In a future version of this cPP TLS v1.2 will be required for all TOEs.*

**FCS_TLSC_EXT.1.2** The TSF shall verify that the presented identifier matches [selection: the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv6 address in the CN or SAN, IPv4 address in SAN, IPv6 address in the SAN, the identifier per RFC 5280 Appendix A using [selection: id-at-commonName, id-at-countryName, id-at-dnQualifier, id-at-generationQualifier, id-at-givenName, id-at-initials, id-at-localityName, id-at-name, id-at-organizationalUnitName, id-at-organizationName, id-at-pseudonym, id-at-serialNumber, id-at-stateOrProvinceName, id-at-surname, id-at-title] and no other attribute types].

***Application Note:***

*Where TLS is used for connections to/from non-TOE entities (relevant to FTP_ITC and FTP_TRP), the ST author shall select RFC 6125. If RFC 5280 is selected, the selection is completed by listing the AttributeType (e.g. 'id-at-serialNumber') as defined in RFC 5280 Appendix A. The selection should only list those attributes that are significant (i.e. those which are used by the client for reference identifier matching), though the Subject field (DN) may contain other attribute types that are not significant for the purpose of reference identifier matching. In the TSS the ST author describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The ST author selects "the reference identifier per RFC 6125 section 6" for TOEs that support FQDN, SRV, and URI identifiers.*

*The ST author selects "IPv4…" and/or "IPv6…" based on the IP versions the TOE supports. The ST author selects "CN or SAN" when IP addresses are supported in the "CN" or "SAN" when the TOE mandates the presence of the SAN. When "CN or SAN" is selected, the TOE only checks the CN when the certificate does not contain the SAN extension.*

*The rules for verification of identity are described in Section 6 of RFC 6125. Additionally, IP address identifiers may be supported in the SAN or CN. The reference identifier is established by the Administrator (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain or IP address and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.*

*The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name may be supported for the purposes of backwards compatibility. When the SAN extension is present in a certificate, the CN must be ignored.*

*Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards and the TOE supports wildcard, the client must follow the best practices regarding matching; these best practices are captured in the evaluation activity. The*

*exception being, the use of wildcards is not supported when using IP address as the reference identifier*

**FCS_TLSC_EXT.1.3** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [selection:

- Not implement any administrator override mechanism
- require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate

].

***Application Note:***

*'Revocation status' refers to an OCSP or CRL response that indicates the presented certificate is invalid. Inability to make a connection to determine validity shall be handled as specified in FIA_X509_EXT.2.2. If the revocation status of a certificate received by the TOE is ambiguous (e.g. 'unknown'), this should be treated similar to the situation where no connection could be established to the revocation server and the option 'determine the revocation status' could be chosen for this.*

*The purpose of the explicit selection in the SFR is to prevent the TOE providing an override mechanism for situations other than specified in the selection (e.g. one or more certificates in the certification path have been revoked and this status is known to the TOE).*

*If TLS is selected in FTP_ITC, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev.*

**FCS_TLSC_EXT.1.4** The TSF shall [selection: not present the Supported Elliptic Curves/Supported Groups Extension, present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: [selection: secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups] in the Client Hello.

***Application Note:***

*If ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, then "not present the Support Elliptic Curves/Supported Groups Extension" should be selected.*

*This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.*

*If ciphersuites with DHE key agreement were selected FCS_TLSC_EXT.1.1 and the TOE supports TLS FFC groups (e.g. ffdhe2048), this extension is required. This extension is not required if the TOE only supports non-TLS FFC groups (e.g. Group 14).*

#### A.2.2.2. FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication

**FCS_TLSS_EXT.1.1** The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] and

reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [selection: *select supported ciphersuites from List 1*] and no other ciphersuites.

***Application Note:***

*The ciphersuites to be tested in the evaluated configuration are limited by this requirement and must be selected from the ciphersuites defined in List 1. The ST author should select the optional ciphersuites that are supported. Even though RFC 5246 mandates implementation of specific ciphers, there is no requirement to implement TLS_RSA_WITH_AES_128_CBC_SHA in order to claim conformance to this cPP.*

*These requirements will be revisited as new TLS versions are standardized by the IETF.*

*In a future version of this cPP TLS v1.2 will be required for all TOEs.*

**FCS_TLSS_EXT.1.2** The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [selection: TLS 1.1, TLS 1.2, none].

***Application Note:***

*All SSL versions and TLS v1.0 are denied. Any TLS versions not selected in FCS_TLSS_EXT.1.1 should be selected here. (If 'none' is the selection for this element then the ST author may omit the words "and none".)*

**FCS_TLSS_EXT.1.3** The TSF shall perform key establishment for TLS using [selection: RSA with key size [selection: 2048 bits, 3072 bits, 4096 bits], Diffie-Hellman parameters with size [selection: 2048 bits, 3072 bits, 4096 bits, 6144 bits, 8192 bits], Diffie-Hellman groups [selection: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, no other groups], ECDHE curves [selection: secp256r1, secp384r1, secp521r1] and no other curves]].

***Application Note:***

*The appropriate options shall be selected in the ST according to the key establishment options supported by the TOE. FMT_SMF.1 requires the configuration of the key agreement parameters to establish the security strength of the TLS connection.*

**FCS_TLSS_EXT.1.4** The TSF shall support [selection: no session resumption or session tickets, session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), session resumption based on session tickets according to RFC 5077].

***Application Note:***

*If the TOE does not support session resumption or session tickets, select 'no session resumption or session tickets'. If the TOE supports session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), select 'session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2)'. If the TOE supports session resumption based on session tickets according to RFC 5077, select 'session resumption based on session tickets according to RFC 5077'.*

### A.2.3. FCS_SSHC_EXT & FCS_SSHS_EXT SSH Protocol

SSH is not a required component of this cPP. If a TOE implements SSH, a corresponding selection in

FTP_ITC.1, FTP_TRP.1/Admin, and/or FTP_TRP.1/NonAdmin should have been made that defines what the SSH protocol is implemented to protect.

A TOE may act as the client or the server in an SSH session. The requirement has been separated into SSH Client (FCS_SSHC_EXT) and SSH Server (FCS_SSHS_EXT) requirements to allow for these differences.

#### A.2.3.1. FCS_SSHC_EXT.1 SSH Client Protocol

(for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

**FCS_SSHC_EXT.1.1** The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [selection: 4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308 section 3.1, 8332].

***Application Note:***

*The following mapping is provided as a guide to ST authors to ensure the appropriate RFC selections are made:*

- RFC 4256 – Select if keyboard-interactive authentication is available
- RFC 4344 – Select if AES-128-CTR or AES-256-CTR modes are available
- RFC 5647 – Select if AEAD_AES_128_GCM or AEAD_AES_256_GCM are available
- RFC 5656 – Select if elliptical curve cryptography is available
- RFC 6187 – Select if X.509 certificates are available for public key algorithms
- RFC 6668 – Select if HMAC-SHA-2 algorithms are available
- RFC 8268 – Select if FFC DH groups with SHA-2 are available
- RFC 8308 Section 3.1 – Select if RFC 8332 is selected
- RFC 8332 – Select if SHA-2 is available with ssh-rsa selection for public key algorithms

*If the negotiated encryption algorithm is one of the aes*-gcm@openssh.com algorithms, then the MAC field is ignored during negotiation and implicitly selects AES-GCM for the MAC. However once negotiated the connection is conformant with RFC 5647 and this should be selected when using aes*-gcm@openssh.com algorithms. aes*-gcm@openssh.com is specified in Section 1.6 of the OpenSSH Protocol Specification (https://cvsweb.openbsd.org/cgi-bin/cvsweb/src/usr.bin/ssh/PROTOCOL?*

*rev=1.31).*

*The ST author selects which of the additional RFCs to which conformance is being claimed. An SSH product can implement additional RFCs, but only those listed in the selection can be claimed as conformant under common criteria. The RFC selections for this requirement need to be consistent with selections in later elements of this Package (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the evaluation activity for this requirement.*

**FCS_SSHC_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, [selection: password-based, no other method].

**FCS_SSHC_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.

***Application Note:***

*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.*

**FCS_SSHC_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [selection: aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr, AEAD_AES_128_GCM, AEAD_AES_256_GCM, aes128-gcm@openssh.com, aes256-gcm@openssh.com].

***Application Note:***

*RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. Corresponding FCS_COP entries are included in the ST for the algorithms selected here.*

**FCS_SSHC_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256] as its public key algorithm(s) and rejects all other public key algorithms.

***Application Note:***

*If x509v3-ssh-rsa, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521 or x509v3-rsa2048-sha256 are selected, then the list of trusted certification authorities must be selected in FCS_SSHC_EXT.1.9 and the FIA_X509_EXT SFRs in Appendix A, Selection-Based Requirements are applicable.*

*It is recommended to configure the TOE to reject presented RSA keys with a key length below 2048 bit.*

*RFC 8332 specifies the use of rsa-sha2-256 or rsa-sha2-512 in SSH.*

*Public-key or certificate-based client authentication within the SSH protocol is based on demonstrated possession of a private key matching a public key associated with a given authorized account on a system.*

*If x509v3-based authentication is claimed, the ST shall also include the appropriate FIA_X509_EXT SFRs.*

*SSH client implementations that claim to support x509v3-based public key authentication algorithms are expected to be able to parse server certificates that comply with RFC 6187 Section 4 recommendations.*

**FCS_SSHC_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM, implicit] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

***Application Note:***

*RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH._

*The ST author selects "implicit" when, and only when, aes*-gcm@openssh.com is selected as an encryption algorithm. When aes*-gcm@openssh.com is negotiated as the encryption algorithm, the MAC algorithm field is ignored and GCM is implicitly used as the MAC. "implicit" is not an SSH algorithm identifier and will not be seen on the wire; however, the negotiated MAC might be decoded as "implicit".*

**FCS_SSHC_EXT.1.7** The TSF shall ensure that [selection: diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, ecdh-sha2-nistp256] and [selection: diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group17-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.

**FCS_SSHC_EXT.1.8** The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

***Application Note:***

*This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold, the encrypted traffic per encryption key needs to be counted. It is also acceptable to count the totally transmitted data per encryption key, the total encrypted traffic for incoming and outgoing data or the total transmitted incoming and outgoing data because the encrypted traffic per encryption key will always be lower or*

*equal to the other options. The rekey requirement applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.*

*It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.*

*For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.*

**FCS_SSHC_EXT.1.9** The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key and [selection: a list of trusted certification authorities, no other methods] as described in RFC 4251 section 4.1.

***Application Note:***

*The list of trusted certification authorities can only be selected if x509v3 -ssh-rsa, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521 or x509v3-rsa2048-sha256 are selected in FCS_SSHC_EXT.1.5.*

#### A.2.3.2. FCS_SSHS_EXT.1 SSH Server Protocol

**FCS_SSHS_EXT.1.1** The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [selection: 4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308 section 3.1, 8332].

***Application Note:***

*The mapping provided in the Section A.2.3.1, "FCS_SSHC_EXT.1 SSH Client Protocol" Application Note may be used as a guide here as well to ensure the appropriate RFC selections are made.*

*If the negotiated encryption algorithm is one of the aes*-gcm@openssh.com algorithms, then the MAC field is ignored during negotiation and implicitly selects AES-GCM for the MAC. However once negotiated the connection is conformant with RFC 5647 and this should be selected when using aes*-gcm@openssh.com algorithms. aes*-gcm@openssh.com is specified in Section 1.6 of the OpenSSH Protocol Specification (https://cvsweb.openbsd.org/cgi-bin/cvsweb/src/usr.bin/ssh/PROTOCOL?rev=1.31).*

*The ST author selects which of the additional RFCs to which conformance is being claimed. An SSH product can implement additional RFCs, but only those listed in the selection can be claimed as conformant under common criteria. The RFC selections for this requirement need to be consistent with selections in later elements of this Package (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the evaluation activity for this requirement.*

**FCS_SSHS_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, [selection: password-based, no other method].

***Application Note:***

*If the TOE supports password-based authentication, the option 'password-based' must be selected.If the TOE supports only public key-based authentication, the option 'no other method' must be chosen.*

**FCS_SSHS_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.

***Application Note:***

*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.*

**FCS_SSHS_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [selection: aes128- cbc, aes256-cbc, aes128-ctr, aes256-ctr, AEAD_AES_128_GCM, AEAD_AES_256_GCM, aes128-gcm@openssh.com, aes256-gcm@openssh.com].

***Application Note:***

*RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as encryption algorithms when the same algorithm is being used as the MAC algorithm. Corresponding FCS_COP entries are included in the ST for the algorithms selected here.*

**FCS_SSHS_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256] as its public key algorithm(s) and rejects all other public key algorithms.

***Application Note:***

*If x509v3-ssh-rsa, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521 or x509v3-rsa2048-sha256 are selected, then the FIA_X509_EXT SFRs in Appendix A, Selection-Based Requirements are applicable.*

*It is recommended to configure the TOE to reject presented RSA keys with a key length below 2048 bit. RFC 8332 specifies the use of rsa-sha2-256 or rsa-sha2-512 in SSH.*

*Public-key or certificate-based client authentication within the SSH protocol is based on the demonstrated possession of a private key matching a public key associated with a given authorized account on a system.*

*If x509v3-based authentication is claimed, the ST shall also include the appropriate FIA_X509_EXT SFRs.*

*An SSH server implementation that claims to support x509v3-based public key authentication algorithms is expected to comply with RFC 6187 Section 4 recommendations when identifying itself with an x.509v3 certificate to SSH clients.*

**FCS_SSHS_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM, implicit] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

***Application Note:***

*RFC 5647 specifies the use of the AEAD_AES_128_GCM and AEAD_AES_256_GCM algorithms in SSH. As described in RFC 5647, AEAD_AES_128_GCM and AEAD_AES_256_GCM can only be chosen as MAC algorithms when the same algorithm is being used as the encryption algorithm. RFC 6668 specifies the use of the sha2 algorithms in SSH.*

*The ST author selects "implicit" when, and only when, aes*-gcm@openssh.com is selected as an encryption algorithm. When aes*-gcm@openssh.com is negotiated as the encryption algorithm, the MAC algorithm field is ignored and GCM is implicitly used as the MAC. "implicit" is not an SSH algorithm identifier and will not be seen on the wire; however, the negotiated MAC might be decoded as "implicit".*

**FCS_SSHS_EXT.1.7** The TSF shall ensure that [selection: diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, ecdh-sha2-nistp256] and [selection: diffie-hellman-group14- sha256, diffie-hellman-group16-sha512, diffie-hellman-group17-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.

**FCS_SSHS_EXT.1.8** The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

***Application Note:***

*This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold, the encrypted traffic per encryption key needs to be counted. It is also accep table to count the totally transmitted data per encryption key, the total encrypted traffic for incoming and outgoing data or the total transmitted incoming and outgoing data because the encrypted traffic per encryption key will always be lower or equal to the other options. The rekey requirement applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.*

*It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.*

*For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.*

### A.2.4. FCS_HTTPS_EXT.1 Extended: HTTPS selected

(selected in FTP_ITC.1.1, FTP_TRP.1.1, for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
[ FCS_TLSC_EXT Extended: TLS Client Protocol, and/or
FCS_TLSS_EXT Extended: TLS Server Protocol ]

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

***Application Note:***

*The ST author must provide enough detail to determine how the implementation is complying with the standard(s) identified; this can be done by additional detail in the TSS.*

**FCS_HTTPS_EXT.1.2** The TSF shall implement HTTPS using TLS as specified in FCS_TLSS_EXT.1 and/or FCS_TLSC_EXT.1.

**FCS_HTTPS_EXT.1.3** If a peer certificate is presented, the TSF shall [selection: not require client authentication, not establish the connection, request authorization to establish the connection, [assignment: other action]] if the peer certificate is deemed invalid.

***Application Note:***

If HTTPS is selected in FTP_ITC.1, FTP_TRP.1/Admin and/or FTP_TRP.1/NonAdmin then validity is determined by the identifier verification, certification path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev.

## A.2.5. FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

(selected with FCS_IPSEC_EXT.1.4, for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)
FTP_ITC.1 or FTP_ITC.2 or FCS_CKM.1
FCS_CKM.4

**FCS_COP.1.1/KeyedHash** The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [selection: *HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, implicit*] and cryptographic key sizes [*assignment: key size (in bits) used in HMAC*] **and message digest sizes [selection: 160, 256, 384, 512] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

***Application Note:***

*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. Select 'implicit' in cases where keyed-hash message authentication is done implicitly (e.g. SSH using AES in GCM mode).*

## A.2.6. FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition

(selected with FCS_IPSEC_EXT.1.4 for O.COMMS_PROTECTION)
**Hierarchical to:**
No other components.

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

***Application Note:***

*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.*

*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.*

*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.*

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

**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 ")").

**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*].

***Application Note:***

*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.*

*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".*

## A.2.7. FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol

Datagram TLS (DTLS) is not a required component of the cPP. If a TOE implements DTLS, a corresponding selection in FTP_ITC.1, FTP_TRP.1/Admin, and/or FTP_TRP.1/NonAdmin should be made to define what the DTLS protocol is implemented to protect. If a corresponding option to support DTLS has been selected in at least one of the SFRs named above, the corresponding selection-based DTLS-related SFRs should be added to the ST from Section A.2.7, "FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol" (i.e. FCS_DTLSC_EXT.1 and/or FCS_DTLSS_EXT.1).

The support for mutual authentication is optional when using DTLS. If a TOE implements DTLS with mutual authentication the corresponding optional SFRs should be added to the ST from Section C.3.1, "FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol" (i.e. FCS_DTLSC_EXT.2 and/or FCS_DTLSS_EXT.2) in addition to the corresponding SFRs from Section A.2.7, "FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol".

The decision whether to include the support for protocol-level mutual authentication in the scope of the evaluation is regarded as part of the TOE boundary definition. These SFRs can be included in a conforming ST at the discretion of the ST author, even if the conformance statement of the cPP requires exact conformance. It is not mandatory to implement mutually authenticated DTLS in order to conform to this cPP.

A TOE may act as the client, the server, or both in DTLS sessions. The requirement has been separated into DTLS Client (FCS_DTLSC_EXT) and DTLS Server (FCS_DTLSS_EXT) requirements to allow for these differences.

If the TOE acts as the client during the claimed DTLS sessions, the ST author should claim the corresponding FCS_DTLSC_EXT requirements.

To ensure audit requirements are properly met, a DTLS receiver may need to monitor the DTLS connection state at the application layer. When no data is received from a DTLS connection for a long time (where the application decides what 'long' means), the receiver should send a close_notify alert message and close the connection.

If the TOE acts as the server during the claimed DTLS sessions, the ST author should claim the corresponding FCS_DTLSS_EXT requirements. In this case the TOE needs to claim at least the FCS_DTLSS_EXT.1 requirements in Section A.2.7, "FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol"

(no support for mutual authentication). If the TOE acts as DTLS server and in addition also supports mutual authentication, the FCS_DTLSS_EXT.2 requirements in Section C.3.2, "FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol" need to be claimed in addition. If the TOE acts as both a client and server during the claimed DTLS sessions, the ST author should claim the corresponding FCS_DTLSC_EXT and FCS_DTLSS_EXT requirements.

#### A.2.7.1. FCS_DTLSC_EXT.1 DTLS Client Protocol Without Mutual Authentication

(for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_DTLSC_EXT.1.1** The TSF shall implement [selection: DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)] supporting the following ciphersuites: [selection: select supported ciphersuites from List 1] and no other ciphersuites.].

***Application Note:***

*The ciphersuites to be tested in the evaluated configuration are limited by this requirement and must be selected from the ciphersuites defined in Section A.2.2, "FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol". The ST author should select the ciphersuites that are supported. Even though RFC 5246 and RFC 6347 mandate implementation of specific ciphers, there is no requirement to implement TLS_RSA_WITH_AES_128_CBC_SHA in order to claim conformance to this cPP.*

*These requirements will be revisited as new DTLS versions are standardized by the IETF.*

*In a future version of this cPP DTLS v1.2 will be required for all TOEs.*

*FCS_DTLSC_EXT.1 should only be used if the TOE transmits application-layer data to an external entity using a trusted channel provided by DTLS without receiving application data that needs to be protected.*

**FCS_DTLSC_EXT.1.2** The TSF shall verify that the presented identifier matches [selection: the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv6 address in the CN or SAN, IPv4 address in SAN, IPv6 address in the SAN, the identifier per RFC 5280 Appendix A using [selection: id-at-commonName, id-at-countryName, id-at-dnQualifier, id-at-generationQualifier, id-at-givenName, id-at-initials, id-at-localityName, id-at-name, id-at-organizationalUnitName, id-at-

organizationName, id-at-pseudonym, id-at-serialNumber, id-at-stateOrProvinceName, id-at-surname, id-at-title] and no other attribute types].

***Application Note:***

*Where DTLS is used for connections to or from non-TOE entities (relevant to FTP_ITC and FTP_TRP) the ST author shall select RFC 6125. If RFC 5280 is selected, the selection is completed by listing the AttributeType (e.g. 'id-at-serialNumber') as defined in RFC 5280 Appendix A. The selection should only list those attributes that are significant (i.e. those which are used by the client for reference identifier matching), though the Subject field (DN) may contain other attribute types that are not significant for the purpose of reference identifier matching. In the TSS, the ST author describes which attribute type, or combination of attributes types, are used by the client to match the presented identifier with the configured identifier. The ST author selects "the reference identifier per RFC 6125 section 6" for TOEs that support FQDN, SRV, and URI identifiers.*

*The ST author selects "IPv4…" and/or "IPv6…" based on the IP versions the TOE supports. The ST author selects "CN or SAN" when IP addresses are supported in the "CN" or "SAN" when the TOE mandates the presence of the SAN. When "CN or SAN" is selected, the TOE only checks the CN when the certificate does not contain the SAN extension.*

*The rules for verification of identity are described in Section 6 of RFC 6125. Additionally, IP address identifiers may be supported in the SAN or CN. The reference identifier is established by the Administrator (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain or IP address and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.*

*The preferred method for verification is the Subject Alternative Name using DNS names, URI names, or Service Names. Verification using the Common Name may be supported for the purposes of backwards compatibility. When the SAN extension is present in a certificate, the CN must be ignored.*

*Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards and the TOE supports wildcard, the client must follow the best practices regarding matching; these best practices are captured in the evaluation activity. The exception being, the use of wildcards is not supported when using IP address as the reference identifier.*

**FCS_DTLSC_EXT.1.3** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [selection:

- Not implement any administrator override mechanism
- require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate

].

***Application Note:***

*'Revocation status' refers to an OCSP or CRL response that indicates the presented certificate is invalid. Inability to make a connection to determine validity shall be handled as specified in FIA_X509_EXT.2.2. If the revocation status of a certificate received by the TOE is ambiguous (e.g. 'unknown'), this should be treated similar to the situation where no connection could be established to the revocation server and the option 'determine the revocation status' could be chosen for this.*

*The purpose of the explicit selection in the SFR is to prevent the TOE from providing an override mechanism for situations other than specified in the selection (e.g. one or more certificates in the certification path have been revoked and this status is known to the TOE).*

*If DTLS is selected in FTP_ITC then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev.*

**FCS_DTLSC_EXT.1.4** The TSF shall [selection: not present the Supported Elliptic Curves/Supported Groups Extension, present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: [selection: secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups] in the Client Hello.

***Application Note:***

*If ciphersuites with elliptic curves were selected in FCS_DTLSC_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_DTLSC_EXT.1.1, then "not present the Supported Elliptic Curves Extension" should be selected.*

*This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.*

*If ciphersuites with DHE key agreement were selected FCS_DTLSC_EXT.1.1 and the TOE supports TLS FFC groups (e.g. ffdhe2048), this extension is required. This extension is not required if the TOE only supports non-TLS FFC groups (e.g. Group 14).*

#### A.2.7.2. FCS_DTLSS_EXT.1 DTLS Server Protocol Without Mutual Authentication

**FCS_DTLSS_EXT.1.1** The TSF shall implement [selection: DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)] supporting the following ciphersuites: [selection: select supported ciphersuites from List 1] and no other ciphersuites.].

***Application Note:***

*The ciphersuites to be tested in the evaluated configuration are limited by this requirement and must be selected from the ciphersuites defined in Section A.2.2, "FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol". The ST author should select the ciphersuites that are supported. Even though RFC 5246 and RFC 6347 mandate implementation of specific ciphers, there is no requirement to implement TLS_RSA_WITH_AES_128_CBC_SHA in order to claim conformance to this cPP.*

*These requirements will be revisited as new DTLS versions are standardized by the IETF.*

*In a future version of this cPP DTLS v1.2 will be required for all TOEs.*

**FCS_DTLSS_EXT.1.2** The TSF shall deny connections from clients requesting *none*.

***Application Note:***

*This version of the cPP does not require the TOE to deny DTLS v1.0. In a future version of this cPP DTLS v1.0 will be required to be denied for all TOEs.*

**FCS_DTLSS_EXT.1.3** The TSF shall not proceed with a connection handshake attempt if the DTLS Client fails validation.

***Application Note:***

*The process to validate the DTLS client is specified in section 4.2.1 of RFC 6347 (DTLS 1.2) and RFC 4347 (DTLS 1.0). The TOE validates the DTLS client during Connection Establishment (Handshaking) and prior to the TSF sending a Server Hello message. After receiving a ClientHello, the DTLS Server sends a HelloVerifyRequest along with a cookie. The cookie is a signed message using the keyed hash function specified in FCS_COP.1/KeyedHash. The DTLS Client then sends another ClientHello with the cookie attached. If the DTLS server successfully verifies the signed cookie, the Client is not using a spoofed IP address.*

**FCS_DTLSS_EXT.1.4** The TSF shall perform key establishment for TLS using [selection: RSA with key size [selection: 2048 bits, 3072 bits, 4096 bits], Diffie-Hellman parameters with size [selection: 2048 bits, 3072 bits, 4096 bits, 6144 bits, 8192 bits], Diffie-Hellman groups [selection: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, no other groups], ECDHE curves [selection: secp256r1, secp384r1, secp521r1] and no other curves].

***Application Note:***

*The appropriate options shall be selected in the ST according to the key establishment options supported by the TOE. FMT_SMF.1 requires the configuration of the key agreement parameters to establish the security strength of the DTLS connection.*

**FCS_DTLSS_EXT.1.5** The TSF shall [selection: terminate the DTLS session, silently discard the record] if a message received contains an invalid MAC.

***Application Note:***

*The Message Authentication Code (MAC) is negotiated during DTLS handshake phase and is used to protect integrity of messages received from the sender during DTLS data exchange. If MAC verification fails, the session must be terminated, or the record must be silently discarded.*

**FCS_DTLSS_EXT.1.6** The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received.
- DTLS records too old to fit in the sliding window.

***Application Note:***

*Replay Detection is described in section 4.1.2.6 of DTLS 1.2 (RFC 6347) and section 4.1.2.5 of DTLS 1.0*

*(RFC 4347). For each received record, the receiver verifies the record contains a sequence number that is within the sliding receive window and does not duplicate the sequence number of any other record received during the session. "Silently Discard" means the TOE discards the packet without responding.*

**FCS_DTLSS_EXT.1.7** The TSF shall support [selection: no session resumption or session tickets, session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), session resumption based on session tickets according to RFC 5077].

***Application Note:***

*If the TOE does not support session resumption or session tickets, select 'no session resumption or session tickets'. If the TOE supports session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), select 'session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2)'. If the TOE supports session resumption based on session tickets according to RFC 5077, select 'session resumption based on session tickets according to RFC 5077'.*

# A.3. Passphrase-based Key Entry

The SFRs in this section are to be incorporated in the ST to support the optional Passphrase-based Key Entry function.

## A.3.1. FCS_PCC_EXT.1 Extended: Cryptographic Password Construct and Conditioning

(for O.STORAGE_ENCRYPTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components

**Dependencies:**
FCS_COP.1/CMAC Cryptographic Operation (for keyed-hash message authentication)

**FCS_PCC_EXT.1.1** A password used by the TSF 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*] bits that meet the following: [**NIST SP 800-132**].

***Application Note:***

*This SFR is conditionally required if the manual entry of a drive encryption passphrase is supported by the TOE.*

## A.3.2. FCS_KDF_EXT Extended: Cryptographic Key Derivation

(selected in FCS_KYC_EXT.1.1 for O.STORAGE_ENCRYPTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components

**Dependencies:**
FCS_COP.1/CMAC Cryptographic Operation (for keyed-hash message authentication),
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction,
[if selected: FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)]

**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, ISO/IEC 11770-6:2016 [selection: KPF2, KPF3, KPF4]*], using the keyed-hash functions specified in FCS_COP.1/CMAC, such that the output is at least of equivalent security strength (in number of bits) to the BEV or the DEK.

### A.3.3. FCS_COP.1/CMAC Cryptographic Operation (for cipher-based message authentication)

(selected with FCS_PCC_EXT.1, FCS_KDF_EXT.1.1, FPT_SBT_EXT.1.2 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/SKG Cryptographic Key Generation (Symmetric Keys)]
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm),
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**FCS_COP.1.1/CMAC Refinement**: The TSF shall perform cryptographic [**message authentication**] in accordance with a specified cryptographic algorithm [**selection: *HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, CMAC-AES-128, CMAC-AES-192, CMAC-AES-256, CMAC-SEED-128, CMAC-HIGHT-128, CMAC-LEA-128, CMAC-LEA-192, CMAC-LEA-256*] and cryptographic key sizes [assignment: **key size (in bits) used in [selection: HMAC, AES, CMAC]*]** that meet the following: [selection:

- ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”, NIST SP 800-38B,
- ISO/IEC 9797-1:2011, Section 7.6 “MAC Algorithm 5”; [selection:
  - ISO/IEC 18033-3:2010, Section 5.4 “SEED”,
  - ISO/IEC 18033-3:2010, Section 4.5 “HIGHT”,
  - ISO/IEC 29192-2:2019, Section 6.3 “LEA”]

]

***Application Note:***

*If one or more HMAC algorithms are selected, the ST author selects "HMAC" in the second selection and "ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'"in the third selection. For the assignment, the key size [k] 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 and L2 = 256 where L2 ≤ k ≤ L1 for HMAC, and the size is either 128, 192, or 256 bits for CMAC.*

*For the assignment, the key size will fall into a range between 128 and 256.*

### A.3.4. FCS_SNI_EXT.1 Extended: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation)

(selected with FCS_PCC_EXT.1, FCS_KDF_EXT.1.1, for O.STORAGE_ENCRYPTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components

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

**FCS_SNI_EXT.1.1** The TSF shall only use salts that are generated by a RNG as specified in FCS_RBG_EXT.1.

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

**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.

].

***Application Note:***

*This SFR is conditionally required if the manual entry of a drive encryption passphrase is supported by the TOE.*

*The requirement for GCM in FCS_SNI_EXT.1.3 is reflective of the requirements in NIST SP 800-38D, Section 8, Section 8.3.*

# A.4. Identification and Authentication (FIA)

## A.4.1. Authentication using X.509 certificates (Extended – FIA_X509_EXT)

X.509 certificate-based authentication is required if IPsec or TLS communications are claimed for FTP_ITC.1 or FTP_TRP. If SSH client communications are claimed and any x509 algorithms are claimed in FCS_SSHC_EXT.1.5 or FCS_SSHS_EXT.1.5, these SFRs are required. In the case of the TOE only acting as the SSH server or acting as the client, but not claiming any x509 algorithms in FCS_SSHC_EXT.1.5 or FCS_SSHS_EXT.1.5, these SFRs are optional.

Although the functionality in FIA_X509_EXT.1/Rev and FIA_X509_EXT.2 is always required when using X.509 certificate-based authentication, the TOE only needs to be able to generate a Certification Request if the TOE needs to present an X.509 certificate to another endpoint via the TSF for authentication (i.e. if at least one of the following SFRs is included in the ST: FCS_DTLSC_EXT.2, FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2, FCS_IPSEC_EXT.1, FCS_SSHC_EXT.1.5 (applicable only if at least one of the x509v3-* ciphers is selected), FCS_SSHS_EXT.1.5 (applicable only if at least one of the x509v3-* ciphers is selected), FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2).. Therefore FIA_X509_EXT.3 only needs to be added to the ST in this case. If the TOE does not need to present an X.509 certificate to another endpoint via the TSF for authentication (e.g. a client not supporting mutual authentication) the use of FIA_X509_EXT.3 is optional.

#### A.4.1.1. FIA_X509_EXT.1 X.509 Certificate Validation

(for O.COMMS_PROTECTION)
**Hierarchical to:**
No other components

**Dependencies:**
FIA_X509_EXT.2 X.509 Certificate Authentication

**FIA_X509_EXT.1.1/Rev** The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates.**
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:

- *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
- *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
- *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
- *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

**FIA_X509_EXT.1.2/Rev** The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

***Application Note:***

*FIA_X509_EXT.1.1/Rev lists the rules for validating certificates. The ST author selects whether revocation status is verified using OCSP or CRLs. The trusted channel/path protocols may require that certificates are used; this use may require that specific certificate extensions must be present and checked. If the TOE supports functionality that does not use any of the possible values listed in the specific certificate extension, then it is reasonable to process such certificate as the relevant part of the SFR is considered trivially satisfied. However, this does not mean that it is allowable to accept certificates with inappropriate extension values simply because a specific security function is not implemented by the TOE. For example, the TOE should not successfully authenticate a web server that presents an X.509v3 certificate that has extendedKeyUsage set to only OCSPSigning, even if the TOE does not implement OCSP revocation checking. The TOE shall be capable of supporting a minimum path length of three certificates. That is, the TOE shall support a hierarchy comprising of at least a self-signed root CA certificate, a subordinate CA certificate, and a leaf certificate. The chain validation is expected to terminate with a trust anchor. This means the validation can terminate with any trusted CA certificate designated as a trust anchor. This CA certificate must be loaded into the trust store ('certificate store', ' trusted CA Key Store' or similar) managed by the TOE trust store. If the TOE's trust store supports loading of multiple hierarchical CA certificates or certificate chains, the TOE must clearly indicate all certificates that it considers trust anchors.*

*The validation of X.509v3 leaf certificates comprises several steps:*

```
a) A Certificate Revocation Check refers to the process of determining the current
revocation status of an otherwise structurally valid certificate. This must be
performed every time a certificate is used for authentication. This check must be
performed for each certificate in the chain up to, but not including, the trust
anchor. This means that CA certificates that are not trust anchors, and leaf
certificates in the chain, must be checked. It is not required to check the revocation
status of any CA certificate designated a trust anchor, however if such check is
performed it must be handled consistently with how other certificates are checked.
```

```
b) An expiration check must be performed. This check must be conducted for each
certificate in the chain, up to and including the trust anchor.
```

```
c) The continuity of the chain must be checked, showing that the signature on each
certificate that is presented to the TOE is valid and the chain terminates at the
trust anchor.
```

```
d) The presence of relevant extensions in each certificate in the chain such as the
extendedKeyUsage parameters of the leaf certificate must correspond to SFR-relevant
functionality. For example, a peer acting as a web server should have TLS Web Server
Authentication listed as an extendedKeyUsage parameter of its X.509v3 certificate. It
shall be checked that the relevant extensions in each certificate in the chain such as
the extendedKeyUsage parameters of the leaf certificate correspond to the SFR-relevant
functionality they are used with.
```

*It is expected that revocation checking is performed when a certificate is used in an authentication step. It is expected that revocation checking is performed on both leaf and intermediate CA certificates when a leaf certificate is presented to the TOE as part of the certificate chain during authentication. Revocation checking of any CA certificate designated a trust anchor is not required.*

*If the TOE implements mutual authentication or acts as a server, there is no expectation of performing any checks on TOE's own leaf certificate during authentication.*

*FIA_X509_EXT.1.2/Rev applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.*

*The ST author must include FIA_X509_EXT.1/Rev in all instances except when only SSH is selected within FTP_ITC.1, and implementation is limited to public-key authentication that does not rely on X.509 certificates. Additionally, FIA_X509_EXT.1/Rev must also be included if "X.509 Certificate" is selected in FPT_TUD_EXT.1.3.*

#### A.4.1.2. FIA_X509_EXT.2 X.509 Certificate Authentication

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FIA_X509_EXT.1 X.509 Certificate Validation

**FIA_X509_EXT.2.1** The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: *DTLS, HTTPS, IPsec, TLS, SSH, [assignment: other protocols], no protocols*] and [selection: *code signing for system firmware/software updates [assignment: other uses], no additional uses*].

**FIA_X509_EXT.2.2** When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: *allow the Administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

***Application Note:***

*In FIA_X509_EXT.2.1, the ST author's selection includes IPsec, TLS, or HTTPS if these protocols are included in FTP_ITC.1.1. SSH should be included if authentication other than ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and/or ecdsa-sha2-nistp521 is selected in FCS_SSHC_EXT.1.5 or FCS_SSHS_EXT.1.5. The ST author selects "code signing for system firmware/software updates" when "X.509 certificate" is selected in FPT_TUD_EXT.1.3.*

*Often a connection must be established to check the revocation status of a certificate - either to download a CRL or to perform a lookup using OCSP. In FIA_X509_EXT.2.2 the selection is used to describe the behaviour in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate is valid according to all other rules in FIA_X509_EXT.1/Rev, the behaviour indicated in the selection determines the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA_X509_EXT.1/Rev. If the Administrator-configured option is selected by the ST Author, the ST Author also selects the corresponding function in FMT_SMF.1. The selection should be consistent with the validation requirements in FCS_IPSEC_EXT.1.14, FCS_TLSC_EXT.1.3 and FCS_TLSC_EXT.2.3.*

*The ST author must include FIA_X509_EXT.2 in all instances except when only SSH is selected within FTP_ITC.1 and ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and/or ecdsa-sha2-nistp521 authentication is also selected. Additionally, FIA_X509_EXT.2 must also be included if "X.509 certificate" is selected in FPT_TUD_EXT.1.3.*

#### A.4.1.3. FIA_X509_EXT.3 X.509 Certificate Requests

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FIA_X509_EXT.1 X.509 Certificate Validation

Although the functionality in FIA_X509_EXT.1/Rev and FIA_X509_EXT.2 is always required when using X.509 certificate-based authentication, the TOE only needs to be able to generate a Certification Request if the TOE needs to present an X.509 certificate to another endpoint via the TSF for authentication (i.e. if at least one of the following SFRs is included in the ST: FCS_DTLSC_EXT.2, FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2, FCS_IPSEC_EXT.1, FCS_SSHC_EXT.1.5 (applicable only if at least one of the x509v3-* ciphers is selected), FCS_SSHS_EXT.1.5 (applicable only if at least one of the x509v3-* ciphers is selected), FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2). Therefore FIA_X509_EXT.3 only needs to be added to the ST in this case. If the TOE does not need to present an X.509 certificate to another endpoint via the TSF for authentication (e.g. a client not supporting mutual authentication) the use of FIA_X509_EXT.3 is optional. This element must be included in the ST if X.509 certificates are used as part of FTP_ITC.1, FTP_TRP.1/Admin, and/or FTP_TRP.1/NonAdmin where the TOE authenticating itself to external IT entities,

administrators, or distributed components.

**FIA_X509_EXT.3.1** The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: *device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: other information]*].

***Application Note:***

*The public key is the public key portion of the public-private key pair generated by the TOE as specified in FCS_CKM.1.*

**FIA_X509_EXT.3.2** The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

# Appendix B: Conditionally Mandatory Requirements

The following are security functional requirements that are mandatory if the TOE configuration meets the condition(s) specified in Section 1.4.2, “USE CASE 2: Conditionally Mandatory Use Cases”.

## B.1. Confidential Data on Nonvolatile Storage Devices

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

(for O.KEY_MATERIAL, for O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
FCS_KYC_EXT.1 Extended: Key Chaining

**FPT_KYP_EXT.1.1 Refinement:** The TSF shall [selection:

- not store keys in non-volatile memory
- only store keys in non-volatile memory when wrapped, as specified in FCS_COP.1/KeyWrap, or encrypted, as specified in FCS_COP.1/KeyEnc or FCS_COP.1/KeyTransport
- only store keys that meet any one of the following criteria [selection:
  - the key is protected by another key that is not part of the key chain as specified in FCS_KYC_EXT.1,
  - the key will no longer provide access to the encrypted data after initial provisioning,
  - the key is a key split that is combined as specified in FCS_SMC_EXT.1, and the other half of the key split is [selection:
    - wrapped as specified in FCS_COP.1/KeyWrap,

        - encrypted as specified in FCS_COP.1/KeyEnc or FCS_COP.1/KeyTransport,
        - derived and not stored in non-volatile memory],
    - the key is [selection:
        - used to wrap a key as specified in FCS_COP.1/KeyWrap,
        - used to encrypt a key as specified in FCS_COP.1/KeyEnc or FCS_COP.1/KeyTransport]
    - that is already [selection:
        - wrapped as specified in FCS_COP.1/KeyWrap,
        - encrypted as specified in FCS_COP.1/KeyEnc or FCS_COP.1/KeyTransport]
    - the non-volatile memory the key is stored on is located in a protected storage device]].

***Application Note:***

*The keys must be protected from unauthorized access and must not be stored on any nonvolatile storage device without protection. If the keys exist within protected memory that is not user accessible on the TOE or OE, the key can be used to protect the BEV or the DEK only if the key is:*

1. *a key that is protected by another key that is not a part of the key chain or;*
2. *a key split or provides additional layers of wrapping or encryption on keys that have already been protected or*
3. *the nonvolatile memory the key is stored on is located in a protected storage device and the key is protected from unauthorized access.*

*Any one of the criteria in the third selection option 3. in FPT_KYP_EXT.1.1 above is used to protect the initial value of the key chain. If "the key is protected by another key that is not part of the key chain as specified in FCS_KYC_EXT.1" selection option is selected, vendors will need to explain what the other key is that is not in the key chain and how the “other key” is used to protect the key (for example, a public key used for encryption, keys only used within a protected storage device or separate co-processor).*

*An example of another key that is not a part of the key chain is as follows. In a protected storage device or separate co-processor, if the key is generated, stored, used, protected from disclosure, and the key is not exportable as a plaintext key, then the key is considered as not belonging to the key chain.*

*The protected storage device can protect stored data from unauthorized access and the nonvolatile memory in it is not accessible from outside of the TOE. Examples of protected storage devices include Secure Elements (SE), Trusted Platform Modules (TPM), Hardware Security Modules (HSM), Trusted Execution Environments (TEE), Secure Enclave Processors (SEP), and so on.*

### B.1.2. FCS_KYC_EXT.1 Extended: Key Chaining

(for O.STORAGE_ENCRYPTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
[ FCS_COP.1/KeyWrap Cryptographic operation (Key Wrapping),
FCS_SMC_EXT.1 Extended: Submask Combining,
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction,
FCS_COP.1/KeyEnc Cryptographic operation (Key Encryption),
FCS_KDF_EXT.1 Cryptographic Operation (Key Derivation),
and/or
FCS_COP.1/KeyTransport Cryptographic operation (Key Transport) ]

***Application Note:***

*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.*

**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/KeyWrap, key combining as specified in FCS_SMC_EXT.1, key encryption as specified in FCS_COP.1/KeyEnc, key derivation as specified in FCS_KDF_EXT.1, key transport as specified in FCS_COP.1/KeyTransport]] while maintaining an effective strength of [selection: 128 bits, 256 bits].

***Application Note:***

*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).*

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

*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.*

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

### 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/StorageEncryption Cryptographic operation (Data Encryption/Decryption).

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

***Application Note:***

*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.*

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

***Application Note:***

*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.*

*If a vendor makes the selection "use a self-encrypting Nonvolatile Storage Device that is separately CC certified to conform to the FDE EE cPP" in FDP_DSK_EXT.1.1, testing is not required as SED testing is performed within the FDE EE cPP already.*

*The TSS, KMD, and test sections only apply to parts of the TOE which fall under the selection "perform encryption in accordance with FCS_COP.1/StorageEncryption".*

# 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.

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

***Application Note:***

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

# B.3. Network Communications

## B.3.1. FTP_TRP.1/NonAdmin Trusted path (for Non-administrators)

(for O.COMMS_PROTECTION)
**Hierarchical to:**
No other components.

**Dependencies:**
[ FCS_IPSEC_EXT.1 Extended: IPsec selected, or
FCS_TLSC_EXT Extended: TLS Client Protocol and/or FCS_TLSS_EXT Extended: TLS Server Protocol, or
FCS_SSHC_EXT Extended: SSH Client Protocol or FCS_SSHS_EXT Extended: SSH Server Protocol, or
FCS_DTLSC_EXT Extended: DTLS Client Protocol and/or FCS_DTLSS_EXT Extended: Server DTLS Protocol, or
FCS_HTTPS_EXT.1 Extended: HTTPS selected ].

**FTP_TRP.1.1/NonAdmin Refinement:** The TSF shall use [selection, choose at least one of: IPsec, SSH, TLS, DTLS, 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.

**FTP_TRP.1.2/NonAdmin Refinement:** The TSF shall permit [selection: the TSF, remote users] to initiate communication via the trusted path

**FTP_TRP.1.3/NonAdmin Refinement:** The TSF shall require the use of the trusted path for initial user authentication and all remote user actions.

***Application Note:***

*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 Section A.2, "Protected Communications" corresponding to their selection are copied to the ST if not already present.*

# B.4. Authentication

### B.4.1. FIA_AFL.1 Authentication failure handling

(for O.USER_I&A, O.AUTH_FAILURES)
**Hierarchical to:**
No other components.

**Dependencies:**
FIA_UAU.1 Timing of authentication

**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*].

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

***Application Note:***

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

# Appendix C: Optional Requirements

ST authors are free to choose none, some or all SFRs defined in this chapter. Just the fact that a product supports a certain functionality does not mandate to add any SFR defined in this chapter.

## C.1. Image Overwrite

The SFRs in this section are to be incorporated in the ST to support the optional Image Overwrite function.

### C.1.1. FDP_RIP.1/Overwrite Subset residual information protection

(for O.IMAGE_OVERWRITE)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

**FDP_RIP.1.1/Overwrite Refinement:** The TSF shall ensure that any previous information content stored on a [selection: **wear-leveled storage device**, **non-wear-leveled storage device**] of a resource is made unavailable [selection: **by overwriting data**, **by destroying its cryptographic key**] upon the **deallocation of the resource from** the following objects: **D.USER.DOC**.

***Application Note:***

*The timing of overwriting or cryptographic key destruction is when completion or cancellation of a Document Processing job, periodically, or when requested by an authorized administrator. In the TSS the ST author will describe which timing is used.*

# C.2. Purge Data

The SFRs in this section are to be incorporated in the ST to support the optional Purge Data function.

## C.2.1. FDP_RIP.1/Purge Subset residual information protection

(for O.PURGE_DATA)
**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

**FDP_RIP.1.1/Purge 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**.

# C.3. Protected Communications (FCS)

## C.3.1. FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol

Datagram TLS (DTLS) is not a required component of the HCDcPP. If a TOE implements DTLS, a corresponding selection in FTP_ITC.1, FTP_TRP.1/Admin, and/or FTP_TRP.1/NonAdmin should be made to define what the DTLS protocol is implemented to protect. If a corresponding option to support DTLS has been selected in at least one of the SFRs named above, the corresponding selection-based DTLS-related SFRs should be added to the ST from chap. Section A.2.7, “FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol” (i.e. FCS_DTLSC_EXT.1 and/or FCS_DTLSS_EXT.1).

The support for mutual authentication is optional when using DTLS. If a TOE implements DTLS with mutual authentication the corresponding optional SFRs should be added to the ST from Section C.3.1, “FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol” (i.e. FCS_DTLSC_EXT.2 and/or FCS_DTLSS_EXT.2) in addition to the corresponding SFRs from Section A.2.7, “FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol”.

A TOE may act as the client, the server, or both in DTLS sessions. The requirement has been separated into DTLS Client (FCS_DTLSC_EXT) and DTLS Server (FCS_DTLSS_EXT) requirements to allow for these differences.

If the TOE acts as the client during the claimed DTLS sessions, the ST author should claim the corresponding FCS_DTLSC_EXT requirements.

To ensure audit requirements are properly met, a DTLS receiver may need to monitor the DTLS connection state at the application layer. When no data is received from a DTLS connection for a long time (where the application decides what "long" means), the receiver should send a close_notify alert message and close the connection.

If the TOE acts as the server during the claimed DTLS sessions, the ST author should claim the corresponding FCS_DTLSS_EXT requirements. In this case the TOE needs to claim at least the FCS_DTLSS_EXT.1 requirements in Section A.2.7, “FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol” (no support for mutual authentication). If the TOE acts as DTLS server and in addition also supports mutual authentication, the FCS_DTLSS_EXT.2 requirements in Section C.3.1, “FCS_DTLSC_EXT & FCS_DTLSS_EXT DTLS Protocol” also need to be claimed in addition. If the TOE acts as both a client and server during the claimed TLS sessions, the ST author should claim the corresponding FCS_TLSC_EXT and FCS_TLSS_EXT requirements.

#### C.3.1.1. FCS_DTLSC_EXT.2 DTLS Client Support for Mutual Authentication

(for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1/DataEncryption Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FCS_DTLSC_EXT.1 DTLS Client Protocol
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_DTLSC_EXT.2.1** The TSF shall support mutual authentication using X.509v3 certificates.

***Application Note:***

*The use of X.509v3 certificates for DTLS is addressed in FIA_X509_EXT.2.1. This requirement adds that the client must be capable of presenting a certificate to a DTLS server for DTLS mutual authentication.*

**FCS_DTLSC_EXT.2.2** The TSF shall [selection: terminate the DTLS session, silently discard the record] if a message received contains an invalid MAC.

***Application Note:***

*The Message Authentication Code (MAC) is negotiated during the DTLS handshake phase and is used to protect the integrity of messages received from the sender during DTLS data exchange. If MAC verification fails, the session must be terminated, or the record must be silently discarded.*

**FCS_DTLSC_EXT.2.3** The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received.
- DTLS records too old to fit in the sliding window.

***Application Note:***

*Replay Detection is described in section 4.1.2.6 of DTLS 1.2 (RFC 6347) and section 4.1.2.5 of DTLS 1.0 (RFC 4347). For each received record, the receiver verifies the record contains a sequence number that is within the sliding receive window and does not duplicate the sequence number of any other record received during the session.*

*"Silently Discard" means the TOE discards the packet without responding.*

#### C.3.1.2. FCS_DTLSS_EXT.2 DTLS Server Support for Mutual Authentication

**FCS_DTLSS_EXT.2.1** The TSF shall support mutual authentication of DTLS clients using X.509v3 certificates.

***Application Note:***

*The use of X.509v3 certificates for DTLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include support for client-side certificates for DTLS mutual authentication.*

**FCS_DTLSS_EXT.2.2** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [selection:

- Not implement any administrator override mechanism
- require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented client certificate

].

***Application Note:***

*'Revocation status' refers to an OCSP or CRL response that indicates the presented certificate is invalid. Inability to make a connection to determine validity shall be handled as specified in FIA_X509_EXT.2.2.*

*If DTLS is selected in FTP_ITC, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev.*

**FCS_DTLSS_EXT.2.3** The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the client.

***Application Note:***

*The client identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison.*

## C.3.2. FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol

TLS is not a required component of this cPP. If a TOE implements TLS, a corresponding selection in FTP_ITC.1, FTP_TRP.1/Admin, and/or FTP_TRP.1/NonAdmin should be made to define what the TLS protocol is implemented to protect. If a corresponding option to support TLS has been selected in at least one of the SFRs named above, the corresponding selection-based TLS related SFRs should be added to the ST from Section A.2.2, "FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol" (i.e. FCS_TLSC_EXT.1 and/or FCS_TLSS_EXT.1).

The support for mutual authentication is optional when using TLS. If a TOE implements TLS with mutual authentication, the corresponding optional SFRs should be added to the ST from Section C.3.2, "FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol" (i.e. FCS_TLSC_EXT.2 and/or FCS_TLSS_EXT.2) in addition to the corresponding SFRs from Section A.2.2, "FCS_TLSC_EXT & FCS_TLSS_EXT TLS Protocol".

A TOE may act as the client, the server, or both in TLS sessions. The requirement has been separated into TLS Client (FCS_TLSC_EXT) and TLS Server (FCS_TLSS_EXT) requirements to allow for these differences. If the TOE acts as the client during the claimed TLS sessions, the ST author should claim the corresponding FCS_TLSC_EXT requirements. If the TOE acts as the server during the claimed TLS sessions, the ST author should claim the corresponding FCS_TLSS_EXT requirements. If the TOE acts as both a client and server during the claimed TLS sessions, the ST author should claim the corresponding FCS_TLSC_EXT and FCS_TLSS_EXT requirements.

### C.3.2.1. FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

(for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FCS_TLSC_EXT.1 TLS Client Protocol without mutual authentication
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_TLSC_EXT.2.1** The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

***Application Note:***

*The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.*

#### C.3.2.2. FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication

**FCS_TLSS_EXT.2.1** The TSF shall support TLS communication with mutual authentication of TLS clients using X.509v3 certificates.

***Application Note:***

*The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.*

**FCS_TLSS_EXT.2.2** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [selection:

- Not implement any administrator override mechanism
- require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented client certificate

].

***Application Note:***

*The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include support for client-side certificates for TLS mutual authentication. If the revocation status of a certificate received by the TOE is unknown, this should be treated similar to the situation where no connection could be established to the revocation server and the option 'determine the revocation status' could be chosen for this. 'Revocation status' refers to an OCSP or CRL response that indicates the presented certificate is invalid. Inability to make a connection to determine validity shall be handled as specified in FIA_X509_EXT.2.2.*

*The purpose of the explicit selection in the SFR is to prevent the TOE from providing an override mechanism for situations other than specified in the selection (e.g. one or more certificates in the certification path have been revoked and this status is known to the TOE). If TLS is selected in FTP_ITC, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev.*

**FCS_TLSS_EXT.2.3** The TSF shall not establish a trusted channel if the identifier contained in a certificate does not match an expected identifier for the client. If the identifier is a Fully Qualified Domain Name (FQDN), then the TSF shall match the identifiers according to RFC 6125, otherwise the TSF shall parse the identifier from the certificate and match the identifier against the expected identifier of the client as described in the TSS.

***Application Note:***

*If the identifier is not a FQDN, then the TSS shall describe how the identifier is parsed from the certificate and matched.*

*The client identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the FQDN, IP address, username, or email address used by the client, or may be p assed to a directory server for*

*comparison.*

# C.4. Asymmetric Key Generation

The SFR in this section is used if the TOE generates asymmetric key pairs for communications.

## C.4.1. FCS_CKM.1/AKG Cryptographic Key Generation (Asymmetric Keys)

(for O.COMMS_PROTECTION, O.STRONG_CRYPTO)
**Hierarchical to:**
No other components.

**Dependencies:**
[ FCS_CKM.2 Cryptographic Key Establishment (Refinement), or
FCS_COP.1/SigGen Cryptographic Operation (for signature generation/verification) ],
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**FCS_CKM.1.1/AKG** The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [selection:

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*
- *ECC schemes using 'NIST curves' [selection: P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;*
- *FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1*
- *FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [selection: RFC 3526, RFC 7919].*
- *KCDSA-based key establishment schemes using key size of [selection: (2048, 224) bit, (2048, 256) bit]: ISO/IEC 14888-3:2018, (Subclause 6.3) "KCDSA"*
- *EC-KCDSA-based key establishment schemes using NIST curve of [selection: P-224, P-256, B-233, B-283, K-233, K-283]: ISO/IEC 14888-3:2018, (Subclause 6.7) "EC-KCDSA"*

] ~~and specified cryptographic key sizes [assignment: cryptographic key sizes] that meet the following: [assignment: list of standards]~~.

***Application Note:***

*The ST author selects all key generation schemes used for key establishment (including generation of ephemeral keys) and device authentication. When key generation is used for key establishment, the schemes in FCS_CKM.2.1 and selected cryptographic protocols must match the selection. When key generation is used for device authentication, other than SSH-RSA, ECDSA-SHA2-NISTP256, ECDSA-SHA2-NISTP384 and ECDSA-SHA2-NISTP521, the public key is expected to be associated with an X.509v3 certificate.*

*If the TOE acts as a receiver in the key establishment schemes and is not configured to support mutual authentication, the TOE does not need to implement key generation.*

*In a distributed TOE, if the TOE component acts as a receiver in the key establishment scheme, the TOE does not need to implement key generation.*

# Appendix D: Extended Component Definitions

This appendix contains the definitions for the extended requirements that are used in the cPP, including those used in [Consistency Rationale] and Appendix A, *Selection-Based Requirements* .

(Note: formatting conventions for selections and assignments in this chapter are those in [CC2].)

## D.1. (FAU)

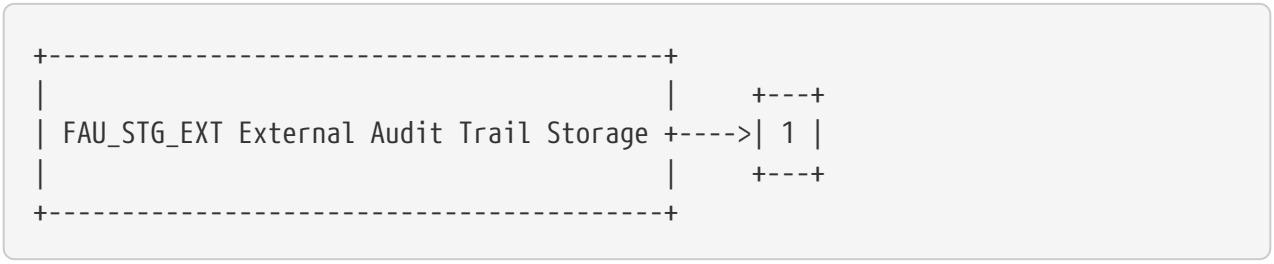
### D.1.1. FAU_STG_EXT Extended: External Audit Trail Storage

**Family Behaviour**

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

**Component Levelling**

*Example 1. Component Levelling*




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

**Management**:

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

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

**Audit**

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.

#### D.1.1.1. 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

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.

**Rationale**

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.

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

# D.2. (FCS)

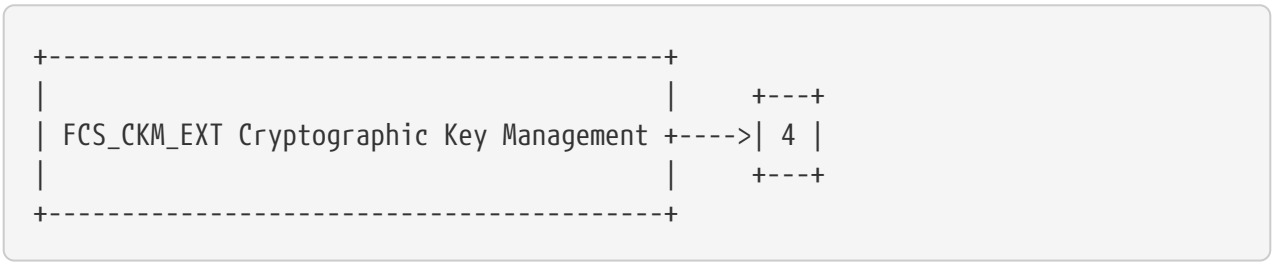
## D.2.1. FCS_CKM_EXT Extended: Cryptographic Key Management

**Family Behaviour**

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

**Component Levelling**

*Example 2. Component Levelling*




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.

**Management**:

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

- There are no management actions foreseen.

**Audit**

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.

#### D.2.1.1. FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction

**Hierarchical to:**
No other components.

**Dependencies:**
[ FCS_CKM.1/AKG Cryptographic Key Generation (Asymmetric Keys), or
FCS_CKM.1/SKG Cryptographic Key Generation (Symmetric Keys)],
FCS_CKM.4 Cryptographic key destruction

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.

**Rationale**

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.

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.

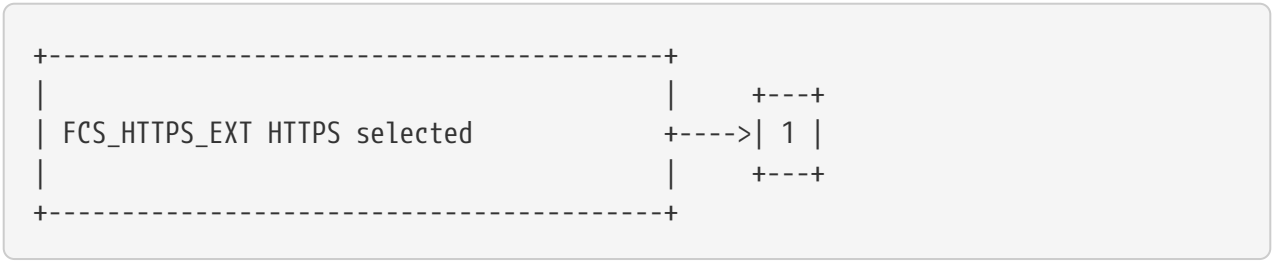
### D.2.2. FCS_HTTPS_EXT Extended: HTTPS selected

**Family Behaviour**

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.

**Component Levelling**

*Example 3. Component Levelling*




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

**Management**

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

- There are no management actions foreseen.

**Audit**

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

- Failure of HTTPS session establishment

#### D.2.2.1. FCS_HTTPS_EXT.1 Extended: HTTPS selected

**Hierarchical to:**
No other components.

**Dependencies:**
[ FCS_TLSC_EXT Extended: TLS Client Protocol, and/or
FCS_TLSS_EXT Extended: TLS Server Protocol ]

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS as specified in FCS_TLSC_EXT.1 and/or FCS_TLSS_EXT.1.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [selection: not require client authentication, not establish the connection, request authorization to establish the connection, [assignment: other action]] if the peer certificate is deemed invalid.

**Rationale**

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.

This extended component protects the communication data using cryptographic algorithms, and it

is therefore placed in the FCS class with a single component.

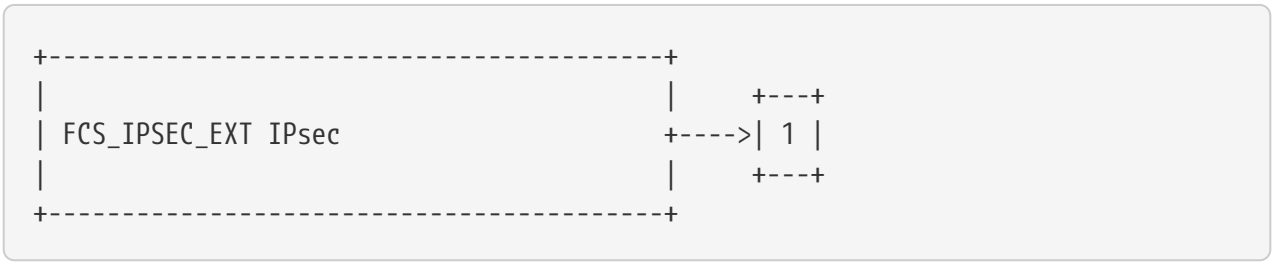
### D.2.3. FCS_IPSEC_EXT Extended: IPsec selected

**Family Behaviour**

This family addresses requirements for protecting communications using IPsec.

**Component Levelling**

*Example 4. Component Levelling*




FCS_IPSEC_EXT.1 IPsec requires that IPsec be implemented as specified.

**Management**

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

- There are no management actions foreseen.

**Audit**

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

#### D.2.3.1. FCS_IPSEC_EXT.1 Extended: IPsec selected

**Hierarchical to:**
No other components.

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

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

**FCS_IPSEC_EXT.1.2** The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched and discards it.

**FCS_IPSEC_EXT.1.3** The TSF shall implement [selection: *transport mode, tunnel mode*].

**FCS_IPSEC_EXT.1.4** The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [selection: AES-CBC-128 (RFC 3602), AES-CBC-192 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-192 (RFC 4106), AES-GCM-256 (RFC 4106),] together with a Secure Hash Algorithm (SHA)-based HMAC [selection: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, no HMAC algorithm].

**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 RFC 5996, section 2.23)], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]].

**FCS_IPSEC_EXT.1.6** The TSF shall ensure the encrypted payload in the [selection: IKEv1, IKEv2] protocol uses the cryptographic algorithms [selection: AES-CBC-128, AES_CBC-192, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-192, AES-GCM-256 (specified in RFC 5282)].

**FCS_IPSEC_EXT.1.7** The TSF shall ensure that [selection:

- IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on [selection:
  - number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range including 24] hours; ];
- IKEv2 SA lifetimes can be configured by a Security Administrator based on [selection:
  - number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range including 24] hours ]

].

**FCS_IPSEC_EXT.1.8** The TSF shall ensure that [selection:

- IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [selection:
  - number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range including 8] hours; ];
- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [selection:
  - number of bytes;
  - length of time, where the time values can be configured within [assignment: integer range

including 8] hours; ]

].

**FCS_IPSEC_EXT.1.9** The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange (“x” in gx mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [assignment: (one or more) number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group] bits.

**FCS_IPSEC_EXT.1.10** The TSF shall generate nonces used in [selection: IKEv1, IKEv2] exchanges of length [selection:

- according to the security strength associated with the negotiated Diffie-Hellman group;
- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash

].

**FCS_IPSEC_EXT.1.11** The TSF shall ensure that IKE protocols implement DH Group(s) [selection:

- [selection: 14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP), 17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526,
- [selection: 19 (256-bit Random ECP), 20 (384-bit Random ECP), 21 (521-bit Random ECP), 24 (2048-bit MODP with 256-bit POS)] according to RFC 5114.

].

**FCS_IPSEC_EXT.1.12** The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1 Phase 1, IKEv2 IKE_SA] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1 Phase 2, IKEv2 CHILD_SA] connection.

**FCS_IPSEC_EXT.1.13** The TSF shall ensure that all IKE protocols perform peer authentication using [selection: RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [selection: Pre-shared Keys, no other method].

**FCS_IPSEC_EXT.1.14** The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: [selection: SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN, CN: IP address, CN: Fully Qualified Domain Name (FQDN), CN: user FQDN, Distinguished Name (DN)] and [selection: no other reference identifier type, [assignment: other supported reference identifier types]].

**Rationale**

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.

This extended component protects the communication data using cryptographic algorithms, and it

is therefore placed in the FCS class with a single component.

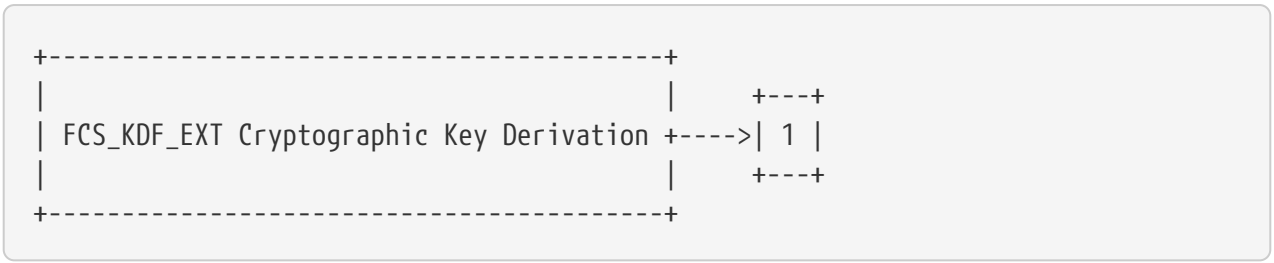
## D.2.4. FCS_KDF_EXT Extended: Cryptographic Key Derivation

**Family Behaviour**

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

**Component Levelling**

*Example 5. Component Levelling*




FCS_KDF_EXT.1 Cryptographic Key Derivation requires the TSF to derive intermediate keys from submasks using the specified hash functions.

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

### D.2.4.1. FCS_KDF_EXT.1 Extended: Cryptographic Key Derivation

**Hierarchical to:**
No other components

**Dependencies:**
FCS_COP.1/CMAC Cryptographic Operation (for keyed-hash message authentication),
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction
[if selected: FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)]

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, ISO/IEC 11770-6:2016 [selection: KPF2, KPF3, KPF4]*], using the keyed-hash functions specified in FCS_COP.1/CMAC, such that the output is at least of equivalent security strength (in number of bits) to the BEV or the DEK.

**Rationale**

The TSF is required to specify the means by which an intermediate key is derived from a specified set of submasks using the specified hash functions.

This extended component protects the Data Encryption Keys using cryptographic algorithms in the maintained key chains, and it is therefore placed in the FCS class with a single component.

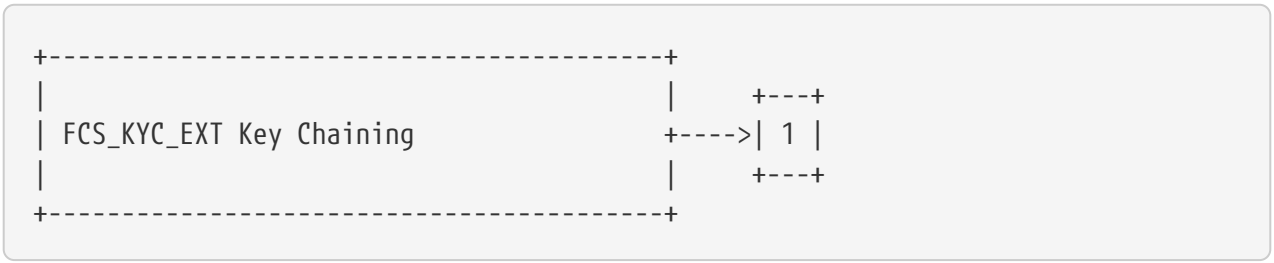
## D.2.5. FCS_KYC_EXT Extended: Cryptographic Operation (Key Chaining)

**Family Behaviour**

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.

**Component Levelling**

*Example 6. Component Levelling*




FCS_KYC_EXT Key Chaining, requires the TSF to maintain a key chain and specifies the characteristics of that chain.

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

### D.2.5.1. FCS_KYC_EXT.1 Extended: Key Chaining

**Hierarchical to:**
No other components.

**Dependencies:**
[ FCS_COP.1/KeyWrap Cryptographic operation (Key Wrapping),
FCS_SMC_EXT.1 Extended: Submask Combining,
FCS_CKM_EXT.4 Extended: Cryptographic Key Material Destruction,
FCS_COP.1/KeyEnc Cryptographic operation (Key Encryption),
FCS_KDF_EXT.1 Cryptographic Operation (Key Derivation),
and/or
FCS_COP.1/KeyTransport Cryptographic operation (Key Transport) ].

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/KeyWrap, key combining as specified in FCS_SMC_EXT.1, key encryption as specified in FCS_COP.1/KeyEnc, key derivation as specified in FCS_KDF_EXT.1, key transport as specified in FCS_COP.1/KeyTransport*]] while maintaining an effective strength of [selection: *128 bits, 256 bits*].

**Rationale**

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.

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

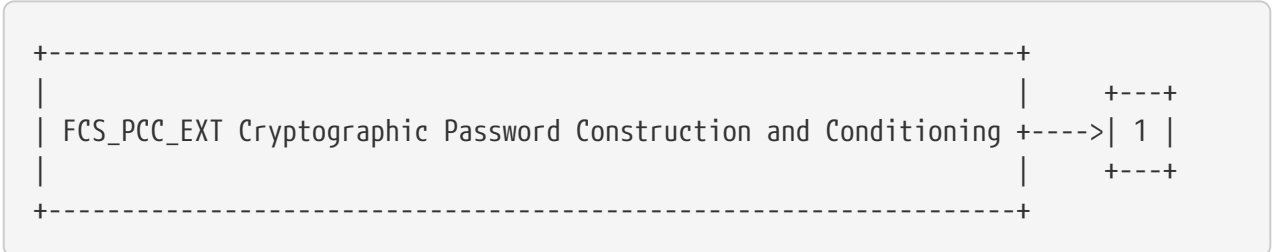
## D.2.6. FCS_PCC_EXT Extended: Cryptographic Password Construction and Conditioning

**Family Behaviour**

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

**Component Levelling**

*Example 7. Component Levelling*



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

**Management**

No specific management functions are identified

**Audit**

There are no auditable events foreseen.

#### D.2.6.1. FCS_PCC_EXT.1 Extended: Cryptographic Password Construction and Conditioning

**Hierarchical to:**
No other components

**Dependencies:**
FCS_COP.1/CMAC Cryptographic Operation (for keyed-hash message authentication)

FCS_PCC_EXT.1.1 A password used by the TSF 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*] bits that meet the following: [assignment: *NIST SP 800-132*].

**Rationale**

The TSF is required to ensure that passwords used to produce the BEV or the DEK are robust (in terms of their composition) and are conditioned to provide an appropriate-length bit string.

This extended component protects the Data Encryption Keys using cryptographic algorithms and Robust BEV in the maintained key chains, and it is therefore placed in the FCS class with a single component.

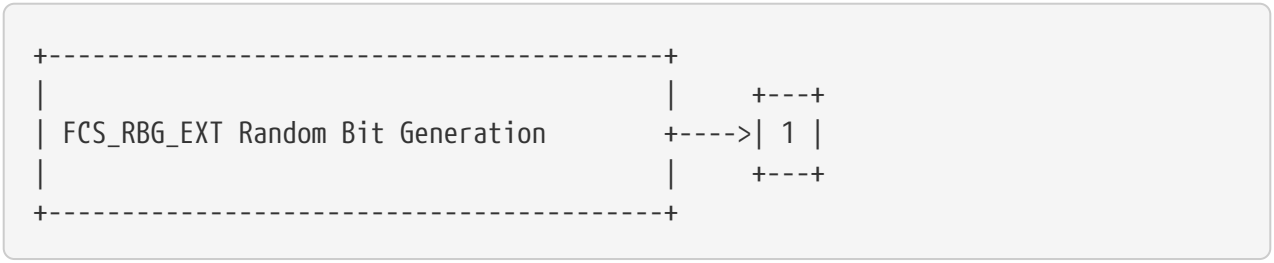
### D.2.7. FCS_RBG_EXT Extended: Cryptographic Operation (Random Bit Generation)

**Family Behaviour**

Components in this family address the requirements for random bit/number generation. This is a new family defined for the FCS class.

**Component Levelling**

*Example 8. Component Levelling*




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.

**Management**

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

- There are no management actions foreseen.

**Audit**

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

- Minimal: failure of the randomization process

#### D.2.7.1. FCS_RBG_EXT.1 Extended: Random Bit Generation

**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

**FCS_RBG_EXT.1.1** The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [selection: *Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG ([selection: AES, SEED, HIGHT, LEA])*].

**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 firmware/software-based sources*] *firmware/software-based noise source*, [*assignment: number of hardware-based sources*] *hardware-based noise source*] with a minimum of [selection: *128 bits, 192 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.

**Rationale**

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.

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

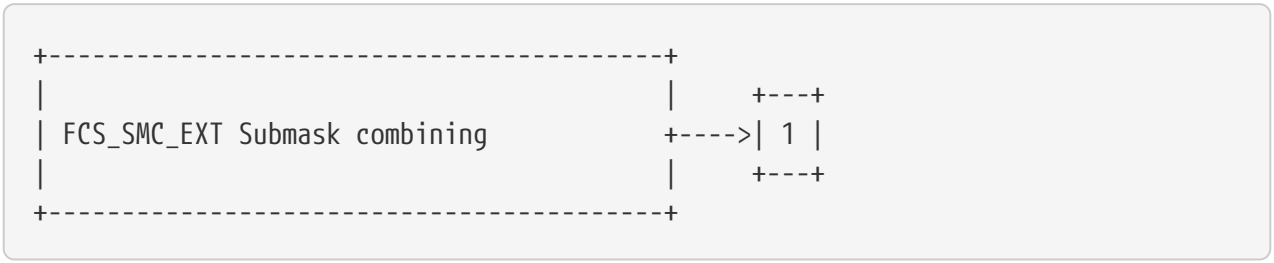
## D.2.8. FCS_SMC_EXT Extended: Submask Combining

**Family Behaviour**

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 or the DEK.

**Component Levelling**

*Example 9. Component Levelling*




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

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

### D.2.8.1. FCS_SMC_EXT.1 Extended: Submask Combining

**Hierarchical to:**
No other components.

**Dependencies:**
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)

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, BEV or DEK.

**Rationale**

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

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

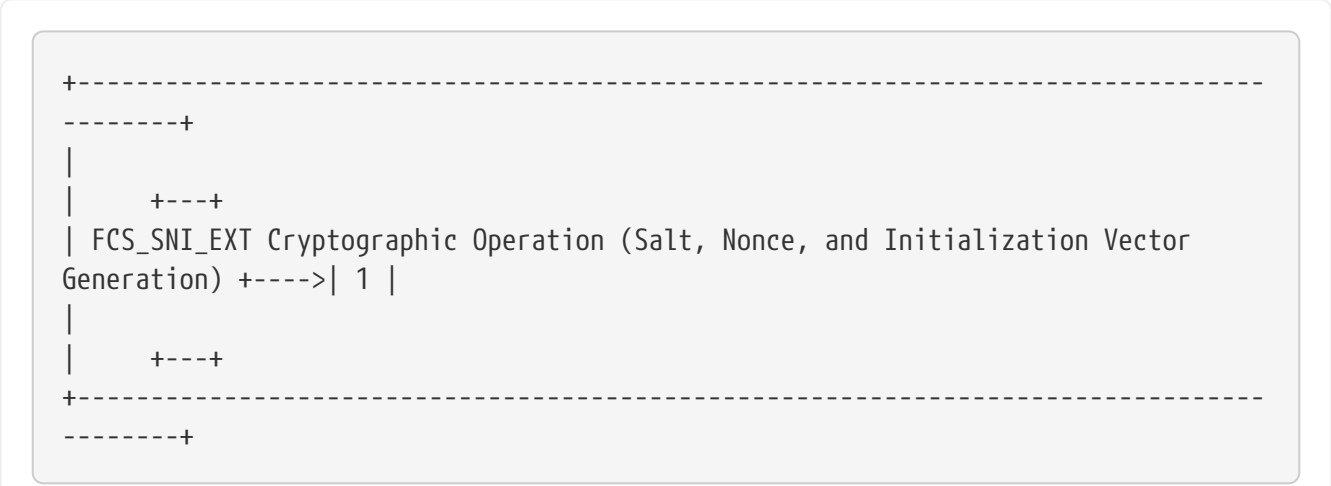
## D.2.9. FCS_SNI_EXT  Extended: Cryptographic Operation (Salt, Nonce, and Initialization Vector Generation

**Family Behaviour**

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

**Component Levelling**

*Example 10. Component Levelling*




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.

**Management**

No specific management functions are identified

**Audit**

There are no auditable events foreseen.

### D.2.9.1. 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)

FCS_SNI_EXT.1.1 The TSF shall only use salts that are generated by a RNG as specified in FCS_RBG_EXT.1.

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

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.

].

**Rationale**

The TSF is required to ensure that the generation of salts, nonces, and IVs to be used by the cryptographic components of the TOE is to be performed in the specified manner.

This extended component protects the communication data and storage data using cryptographic algorithms with specified Salt, Nonce and Initialization Vector Generation, and it is therefore placed in the FCS class with a single component.

## D.2.10. FCS_SSHC_EXT.1 SSH Client

**Family Behaviour**

The component in this family addresses the ability for a client to use SSH to protect data between the client and a server using the SSH protocol.

**Component Levelling** FCS_SSHC_EXT.1 SSH Client requires that the client side of SSH be implemented as specified.

**Management**: FCS_SSHC_EXT.1 The following actions could be considered for the management functions in FMT:

- There are no management activities foreseen.

**Audit**: FCS_SSHC_EXT.1

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

- Failure of SSH session establishment
- SSH session establishment
- SSH session termination

#### D.2.10.1. FCS_SSHC_EXT.1

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

**FCS_SSHC_EXT.1.1** The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [selection: 4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308 section 3.1, 8332].

**FCS_SSHC_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, [selection: password-based, no other method].

**FCS_SSHC_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.

**FCS_SSHC_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [selection: aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr, AEAD_AES_128_GCM, AEAD_AES_256_GCM, aes128-gcm@openssh.com, aes256-gcm@openssh.com].

**FCS_SSHC_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256] as its public key algorithm(s) and rejects all other public key algorithms

**FCS_SSHC_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM, implicit] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS_SSHC_EXT.1.7** The TSF shall ensure that [selection: diffie-hellman-group14-sha1, diffie-

hellman-group15-sha512, ecdh-sha2-nistp256] and [selection: diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group17-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.

**FCS_SSHC_EXT.1.8** The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

**FCS_SSHC_EXT.1.9** The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key and [selection: a list of trusted certification authorities, no other methods] as described in RFC 4251 section 4.1.

## D.2.11. FCS_SSHS_EXT.1 SSH Server Protocol

**Family Behaviour** The component in this family addresses the ability for a server to offer SSH to protect data between a client and the server using the SSH protocol.

**Component Levelling** FCS_SSHS_EXT.1 SSH Server requires that the server side of SSH be implemented as specified.

**Management**: FCS_SSHS_EXT.1 The following actions could be considered for the management functions in FMT:

- There are no management activities foreseen.

**Audit**: FCS_SSHS_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit datageneration is included in the PP/ST:

- Failure of SSH session establishment
- SSH session establishment
- SSH session termination

#### D.2.11.1. FCS_SSHS_EXT.1

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

**FCS_SSHS_EXT.1.1** The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [selection: 4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308 section 3.1, 8332].

**FCS_SSHS_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, [selection: *password-based, no other method*].

**FCS_SSHS_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.

**FCS_SSHS_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [selection: aes128- cbc, aes256-cbc, aes128-ctr, aes256-ctr, AEAD_AES_128_GCM, AEAD_AES_256_GCM, aes128-gcm@openssh.com, aes256-gcm@openssh.com].

**FCS_SSHS_EXT.1.5** The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256] as its public key algorithm(s) and rejects all other public key algorithms.

**FCS_SSHS_EXT.1.6** The TSF shall ensure that the SSH transport implementation uses [selection: hmac-sha1, hmac-sha1-96, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM, implicit] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

**FCS_SSHS_EXT.1.7** The TSF shall ensure that [selection: diffie-hellman-group14-sha1, diffie-hellman-group15-sha512, ecdh-sha2-nistp256] and [selection: diffie-hellman-group14- sha256, diffie-hellman-group16-sha512, diffie-hellman-group17-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521, no other methods] are the only allowed key exchange methods used for the SSH protocol.

**FCS_SSHS_EXT.1.8** The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

### D.2.12. FCS_TLSC_EXT Extended: TLS Client Protocol

**Family Behaviour**

The component in this family addresses the ability for a client to use TLS to protect data between the client and a server using the TLS protocol.

**Component Levelling**

FCS_TLSC_EXT.1 TLS Client requires that the client side of TLS be implemented as specified.

FCS_TLSC_EXT.2 TLS Client requires that the client side of the TLS implementation include mutual authentication.

**Management**: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

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

- There are no management activities foreseen.

**Audit**: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

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

- Failure of TLS session establishment
- TLS session establishment
- TLS session termination

#### D.2.12.1. FCS_TLSC_EXT.1 TLS Client Protocol without Mutual Authentication

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_TLSC_EXT.1.1** The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [selection: *select supported ciphersuites from List 1*] and no other ciphersuites.

**FCS_TLSC_EXT.1.2** The TSF shall verify that the presented identifier matches [selection: the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv6 address in the CN or SAN, IPv4 address in SAN, IPv6 address in the SAN, the identifier per RFC 5280 Appendix A using [selection: id-at-commonName, id-at-countryName, id-at-dnQualifier, id-at-generationQualifier, id-at-givenName, id-at-initials, id-at-localityName, id-at-name, id-at-organizationalUnitName, id-at-organizationName, id-at-pseudonym, id-at-serialNumber, id-at-stateOrProvinceName, id-at-surname, id-at-title] and no other attribute types].

**FCS_TLSC_EXT.1.3** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [selection:

- Not implement any administrator override mechanism
- require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate

].

**FCS_TLSC_EXT.1.4** The TSF shall [selection: not present the Supported Elliptic Curves/Supported Groups Extension, present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: [selection: secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups] in the Client Hello.

#### D.2.12.2. FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FCS_TLSC_EXT.1 TLS Client Protocol without mutual authentication
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_TLSC_EXT.2.1** The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

### D.2.13. FCS_TLSS_EXT Extended: TLS Server Protocol

**Family Behaviour** The component in this family addresses the ability for a server to use TLS to protect data between a client and the server using the TLS protocol.

**Component Levelling**

FCS_TLSS_EXT.1 TLS Server requires that the server side of TLS be implemented as specified.

FCS_TLSS_EXT.2: TLS Server requires the mutual authentication be included in the TLS implementation.

**Management**: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

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

- There are no management activities foreseen.

**Audit**: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

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

- Failure of TLS session establishment
- TLS session establishment
- TLS session termination

#### D.2.13.1. FCS_TLSS_EXT.1 TLS Server Protocol without Mutual Authentication

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_TLSS_EXT.1.1** The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [selection: *select supported ciphersuites from List 1*] and no other ciphersuites.

**FCS_TLSS_EXT.1.2** The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [selection: TLS 1.1, TLS 1.2, none].

**FCS_TLSS_EXT.1.3** The TSF shall perform key establishment for TLS using [selection: RSA with key size [selection: 2048 bits, 3072 bits, 4096 bits], Diffie-Hellman parameters with size [selection: 2048 bits, 3072 bits, 4096 bits, 6144 bits, 8192 bits], Diffie-Hellman groups [selection: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, no other groups], ECDHE curves [selection: secp256r1,

secp384r1, secp521r1] and no other curves]].

**FCS_TLSS_EXT.1.4** The TSF shall support [selection: no session resumption or session tickets, session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), session resumption based on session tickets according to RFC 5077].

#### D.2.13.2. FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FCS_TLSS_EXT.1 TLS Server Protocol without mutual authentication
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_TLSS_EXT.2.1** The TSF shall support TLS communication with mutual authentication of TLS clients using X.509v3 certificates.

**FCS_TLSS_EXT.2.2** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [selection:

- Not implement any administrator override mechanism
- require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented client certificate

].

**FCS_TLSS_EXT.2.3** The TSF shall not establish a trusted channel if the identifier contained in a certificate does not match an expected identifier for the client. If the identifier is a Fully Qualified Domain Name (FQDN), then the TSF shall match the identifiers according to RFC 6125, otherwise the TSF shall parse the identifier from the certificate and match the identifier against the expected identifier of the client as described in the TSS.

### D.2.14. FCS_DTLSC_EXT Extended: DTLS Client Protocol

**Family Behaviour**

The component in this family addresses the ability for a client to use DTLS to protect data between the client and a server using the DTLS protocol.

**Component Levelling**

FCS_DTLSC_EXT.1 DTLS Client requires that the client side of DTLS be implemented as specified.

FCS_DTLSC_EXT.2 DTLS Client requires that the client side of the DTLS implementation include mutual authentication.

**Management**: FCS_DTLSC_EXT.1, FCS_DTLSC_EXT.2

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

- There are no management activities foreseen.

**Audit**: FCS_DTLSC_EXT.1, FCS_DTLSC_EXT.2

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

- Failure of DTLS session establishment
- DTLS session establishment
- DTLS session termination

#### D.2.14.1. FCS_DTLSC_EXT.1 DTLS Client Protocol

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_DTLSC_EXT.1.1** The TSF shall implement [selection: DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)] supporting the following ciphersuites: [selection: *select supported ciphersuites from List 1*] and no other ciphersuites.].

**FCS_DTLSC_EXT.1.2** The TSF shall verify that the presented identifier matches [selection: the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv6 address in the CN or SAN, IPv4 address in SAN, IPv6 address in the SAN, the identifier per RFC 5280 Appendix A using [selection: id-at-commonName, id-at-countryName, id-at-dnQualifier, id-at-generationQualifier, id-at-givenName, id-at-initials, id-at-localityName, id-at-name, id-at-organizationalUnitName, id-at-organizationName, id-at-pseudonym, id-at-serialNumber, id-at-stateOrProvinceName, id-at-

surname, id-at-title] and no other attribute types].

**FCS_DTLSC_EXT.1.3** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [selection:

- Not implement any administrator override mechanism
- require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate

].

**FCS_DTLSC_EXT.1.4** The TSF shall [selection: not present the Supported Elliptic Curves/Supported Groups Extension, present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: [selection: secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups] in the Client Hello.

#### D.2.14.2. FCS_DTLSC_EXT.2 DTLS Client Support for Mutual Authentication

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1/DataEncryption Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FCS_DTLSC_EXT.1 DTLS Client Protocol
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_DTLSC_EXT.2.1** The TSF shall support mutual authentication using X.509v3 certificates.

**FCS_DTLSC_EXT.2.2** The TSF shall [selection: terminate the DTLS session, silently discard the record] if a message received contains an invalid MAC.

**FCS_DTLSC_EXT.2.3** The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received;
- DTLS records too old to fit in the sliding window.

### D.2.15. FCS_DTLSS_EXT Extended: DTLS Server Protocol

**Family Behaviour**

The component in this family addresses the ability for a server to use DTLS to protect data between a client and the server using the DTLS protocol.

**Component Levelling**

FCS_DTLSS_EXT.1 DTLS Server requires that the server side of TLS be implemented as specified.

FCS_DTLSS_EXT.2: DTLS Server requires that mutual authentication be included in the DTLS implementation.

**Management**: FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2

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

- There are no management activities foreseen.

**Audit**: FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2

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

- Failure of DTLS session establishment.
- DTLS session establishment
- DTLS session termination

#### D.2.15.1. FCS_DTLSS_EXT.1 DTLS Server Protocol

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

**FCS_DTLSS_EXT.1.1** The TSF shall implement [selection: DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)] supporting the following ciphersuites: [selection: select supported ciphersuites from List 1] and no other ciphersuites.].

**FCS_DTLSS_EXT.1.2** The TSF shall deny connections from clients requesting *none*.

**FCS_DTLSS_EXT.1.3** The TSF shall not proceed with a connection handshake attempt if the DTLS Client fails validation.

**FCS_DTLSS_EXT.1.4** The TSF shall perform key establishment for TLS using [selection: RSA with key size [selection: 2048 bits, 3072 bits, 4096 bits], Diffie-Hellman parameters with size [selection: 2048 bits, 3072 bits, 4096 bits, 6144 bits, 8192 bits], Diffie-Hellman groups [selection: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, no other groups], ECDHE curves [selection: secp256r1, secp384r1, secp521r1] and no other curves].

**FCS_DTLSS_EXT.1.5** The TSF shall [selection: terminate the DTLS session, silently discard the record] if a message received contains an invalid MAC.

**FCS_DTLSS_EXT.1.6** The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received.
- DTLS Records too old to fit in the sliding window.

**FCS_DTLSS_EXT.1.7** The TSF shall support [selection: no session resumption or session tickets, session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), session resumption based on session tickets according to RFC 5077].

#### D.2.15.2. FCS_DTLSS_EXT.2 DTLS Server Support for Mutual Authentication

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (Data Encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FCS_DTLSS_EXT.1 DTLS Server Protocol

**FCS_DTLSS_EXT.2.1** The TSF shall support mutual authentication of DTLS clients using X.509v3 certificates.

**FCS_DTLSS_EXT.2.2** When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [selection:

- Not implement any administrator override mechanism
- require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented client certificate

].

**FCS_DTLSS_EXT.2.3** The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for

the client.

# D.3. (FDP)

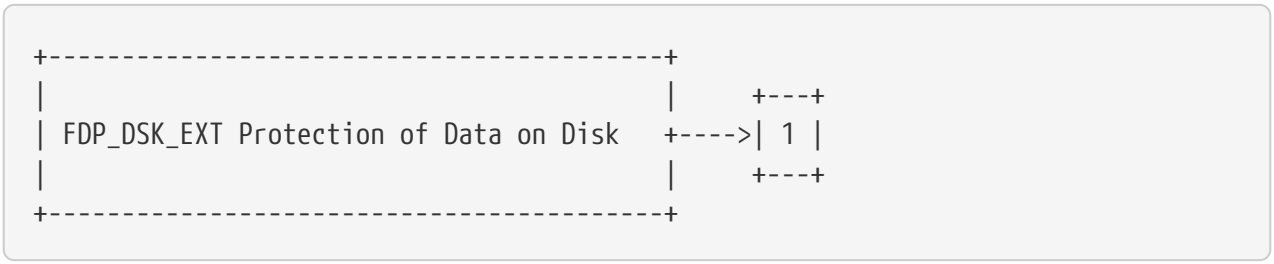
## D.3.1. FDP_DSK_EXT Extended: Protection of Data on Disk

**Family Behaviour**

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

**Component Levelling**

*Example 11. Component Levelling*




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 Nonvolatile Storage Devices in order to avoid storing these data in plaintext on the devices.

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

### D.3.1.1. FDP_DSK_EXT.1 Extended: Protection of Data on Disk

**Hierarchical to:**
No other components.

**Dependencies:**
FCS_COP.1/StorageEncryption Cryptographic operation (Data Encryption/Decryption)

FDP_DSK_EXT.1.1 The TSF shall [selection: *perform encryption in accordance with FCS_COP.1/StorageEncryption, use a self-encrypting Nonvolatile Storage Device that is separately CC*

*certified to conform to the FDE EE cPP*] such that any Nonvolatile Storage Device contains no plaintext User Document Data and no plaintext confidential TSF Data.

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

**Rationale**

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.

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

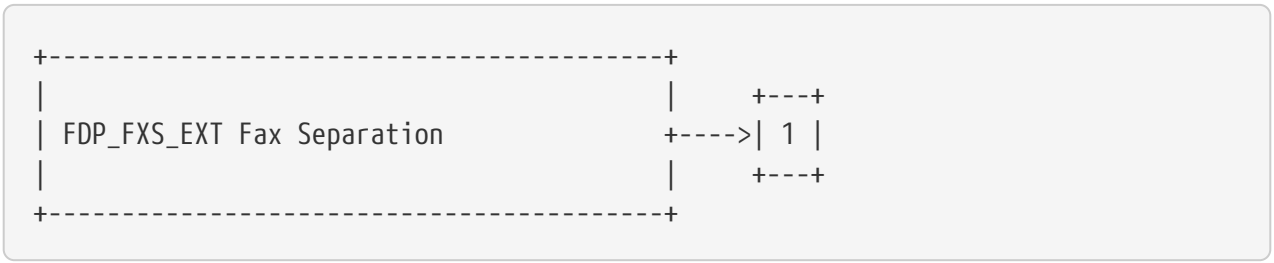
## D.3.2. FDP_FXS_EXT Extended: Fax Separation

**Family Behaviour**

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

**Component Levelling**

*Example 12. Component Levelling*




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.

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

#### D.3.2.1. FDP_FXS_EXT.1 Extended: Fax separation

**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

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

**Rationale**

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.

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

# D.4. (FIA)

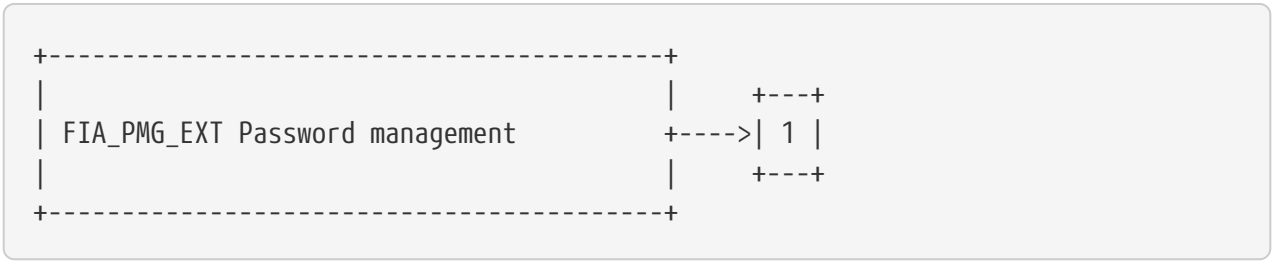
## D.4.1. FIA_PMG_EXT Extended: Password Management

**Family Behaviour**

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.

**Component Levelling**

*Example 13. Component Levelling*




FIA_PMG_EXT.1 Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

#### D.4.1.1. FIA_PMG_EXT.1 Extended: Password management

**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

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.

**Rationale**

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.

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

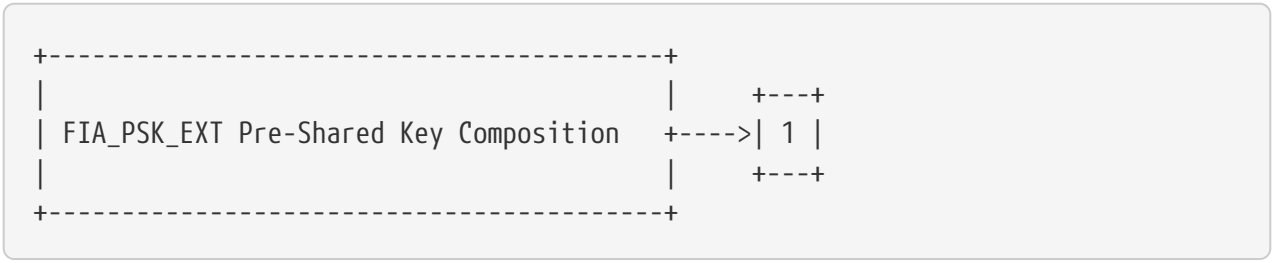
### D.4.2. FIA_PSK_EXT Extended: Pre-Shared Key Composition

**Family Behaviour**

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

**Component Levelling**

*Example 14. Component Levelling*




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

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

#### D.4.2.1. 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).

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

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 “)”).

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*].

**Rationale**

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.

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

## D.4.3. Authentication using X.509 certificates (FIA_X509_EXT)

**Family Behaviour**

This family defines the behaviour, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates for authentication for protocols and integrity verification, and the generation of certificate requests.

**Component Levelling**

FIA_X509_EXT.1 X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.

FIA_X509_EXT.2 X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.

FIA_X509_EXT.3 X509 Certificate Requests, requires the TSF to be able to generate Certificate Request Messages and validate responses.

**Management**: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

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

- Remove imported X.509v3 certificates
- Approve import and removal of X.509v3 certificates
- Initiate certificate requests

**Audit**: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

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

- Minimal: No specific audit requirements are specified.

### D.4.3.1. FIA_X509_EXT.1 X.509 Certificate Validation

**Hierarchical to:**
No other components

**Dependencies:**
FIA_X509_EXT.2 X.509 Certificate Authentication

**FIA_X509_EXT.1.1/Rev** The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates.**
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:

  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

**FIA_X509_EXT.1.2/Rev** The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

#### D.4.3.2. FIA_X509_EXT.2 X509 Certificate Authentication

**Hierarchical to:**
No other components

**Dependencies:**
FIA_X509_EXT.1 X.509 Certificate Validation

**FIA_X509_EXT.2.1** The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: *DTLS, HTTPS, IPsec, TLS, SSH, [assignment: other protocols], no protocols*], and [selection: *code signing for system firmware/software updates [assignment: other uses], no additional uses*].

**FIA_X509_EXT.2.2** When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: *allow the Administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

#### D.4.3.3. FIA_X509_EXT.3 X.509 Certificate Requests

**Hierarchical to:**
No other components

**Dependencies:**
FCS_CKM.1 Cryptographic Key Generation
FIA_X509_EXT.1 X.509 Certificate Validation

**FIA_X509_EXT.3.1** The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: *device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: other information]*].

**FIA_X509_EXT.3.2** The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

# D.5. (FPT)

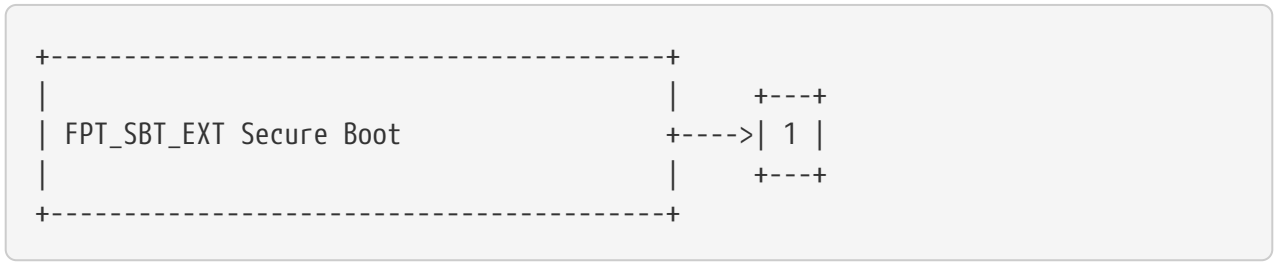
## D.5.1. FPT_SBT_EXT Extended: Secure Boot

**Family Behaviour**

This family addresses the requirements for verifying firmware/software integrity each time that that it is powered on.

**Component Levelling**

*Example 15. Component Levelling*




FPT_SBT_EXT.1 Secure Boot, uses a Root of Trust to confirm the integrity of the device's firmware/software at boot time.

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

#### D.5.1.1. FPT_SBT_EXT.1 Extended: Secure Boot

**Hierarchical to:**
No other components.

**Dependencies:**
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/SigGen Cryptographic Operation (for signature generation/verification)
FCS_COP.1/KeyedHash Cryptographic Operation (for keyed-hash message authentication)
FCS_COP.1/DataEncryption Cryptographic Operation (Symmetric encryption/decryption)
FCS_COP.1/StorageEncryption Cryptographic operation (Data Encryption/Decryption)
FCS_COP.1/CMAC Cryptographic Operation (for keyed-hash message authentication)

**FPT_SBT_EXT.1.1** The TSF shall contain one or more chains of trust with each chain of trust anchored in a Root of Trust that is implemented in immutable code or a HW-based write-protection mechanism.

**FPT_SBT_EXT.1.2** At boot time the TSF shall use the chain(s) of trust to confirm integrity of its firmware/software using a [selection: *hash, digital signature, message authentication*] verification method.

**FPT_SBT_EXT.1.3** The TSF shall [selection: *enter maintenance mode, halt boot process, reboot the device, [assignment: another behavior of TOE]*] in the event of a boot time verification failure so that the corrupted firmware/software isn't executed.

**FPT_SBT_EXT.1.4** Following failure of verification, the TSF shall provide a mechanism to: [selection: *revert to previous TOE image, reinstall TOE image, perform a factory reset, indicate a need to contact vendor support*].

**FPT_SBT_EXT.1.5** The TSF shall contain [selection: *hash data, digital signature data, message authentication code, public key for digital signature, symmetric key for message authentication with confidentiality protection as defined in FPT_SBT_EXT.1.6*] in the Hardware Root of Trust.

**FPT_SBT_EXT.1.6** The TSF shall make the symmetric key accessible only to the Hardware Root of Trust.

**Rationale**

Secure Boot is to verify the integrity of the boot process starting with the hardware-anchored Root of Trust and then verifying each link in the corresponding Chain of Trust to ensure that no corrupted firmware/software is executed.

This extended component verifies the integrity of the Chains of Trusts which are TSF data, and it is therefore placed in the FPT class with a single component.

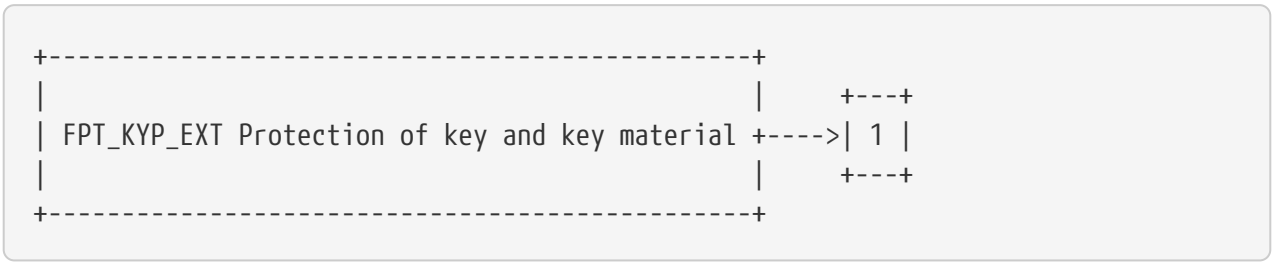
### D.5.2. FPT_KYP_EXT Extended: Protection of Key and Key Material

**Family Behaviour**

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

**Component Levelling**

*Example 16. Component Levelling*




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.

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

#### D.5.2.1. FPT_KYP_EXT.1 Extended: Protection of Key and Key Material

**Hierarchical to:**
No other components.

**Dependencies:**
FCS_KYC_EXT.1 Extended: Key Chaining

FPT_KYP_EXT.1.1 Refinement: The TSF shall [selection:

- not store keys in non-volatile memory
- only store keys in non-volatile memory when wrapped, as specified in FCS_COP.1/KeyWrap, or encrypted, as specified in FCS_COP.1/KeyEnc or FCS_COP.1/KeyTransport
- only store keys that meet any one of the following criteria [selection:

- the key is protected by another key that is not part of the key chain as specified in FCS_KYC_EXT.1,
- the key will no longer provide access to the encrypted data after initial provisioning,
- the key is a key split that is combined as specified in FCS_SMC_EXT.1, and the other half of the key split is [selection:
  - wrapped as specified in FCS_COP.1/KeyWrap,
  - encrypted as specified in FCS_COP.1/KeyEnc or FCS_COP.1/KeyTransport,
  - derived and not stored in non-volatile memory],
- the key is [selection:
  - used to wrap a key as specified in FCS_COP.1/KeyWrap,
  - used to encrypt a key as specified in FCS_COP.1/KeyEnc or FCS_COP.1/KeyTransport]
- that is already [selection:
  - wrapped as specified in FCS_COP.1/KeyWrap,
  - encrypted as specified in FCS_COP.1/KeyEnc or FCS_COP.1/KeyTransport]
- the non-volatile memory the key is stored on is located in a protected storage device]].

**Rationale**

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.

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

## D.5.3. FPT_SKP_EXT Extended: Protection of TSF Data

**Family Behaviour**

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.

**Component Levelling**

*Example 17. Component Levelling*


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.

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

#### D.5.3.1. FPT_SKP_EXT.1 Extended: Protection of TSF Data

**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

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

**Rationale**

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.

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.

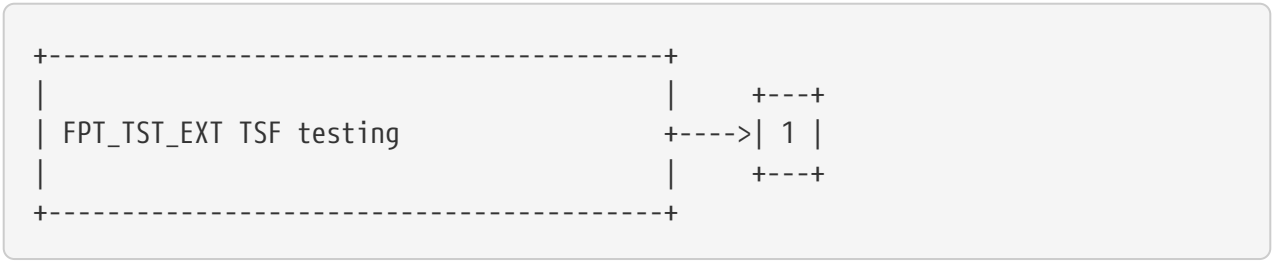
### D.5.4. FPT_TST_EXT Extended: TSF testing

**Family Behaviour**

This family addresses the requirements for self-testing the TSF for selected correct operation.

**Component Levelling**

*Example 18. Component Levelling*




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.

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

#### D.5.4.1. FPT_TST_EXT.1 Extended: TSF testing

**Hierarchical to:**
No other components.

**Dependencies:**
No dependencies.

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.

**Rationale**

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.

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

### D.5.5. FPT_TUD_EXT Extended: Trusted Update

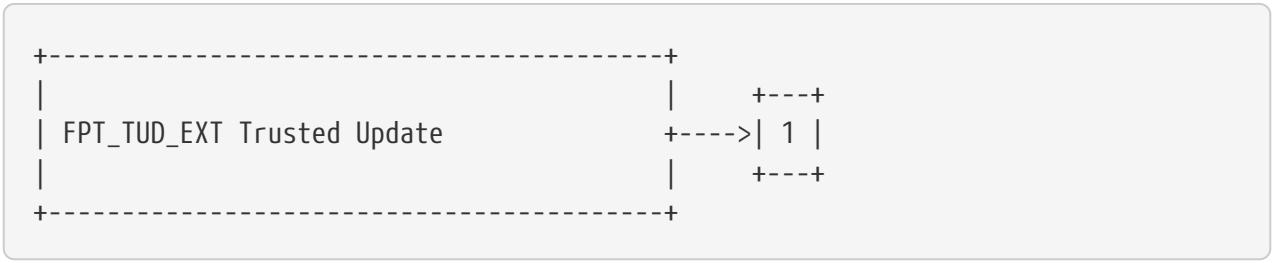
**Family Behaviour**

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.

**Component Levelling**

*Example 19. Component Levelling*




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

**Management**

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

- There are no management actions foreseen.

**Audit**

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.

#### D.5.5.1. FPT_TUD_EXT.1 Trusted Update

**Hierarchical to:**
No other components.

**Dependencies:**
[ FCS_COP.1/SigGen Cryptographic Operation (for signature generation/verification), FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) ].

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

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

FPT_TUD_EXT.1.3 The TSF shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [selection: published hash, no other functions] prior to installing those updates.

**Rationale**

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.

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

# Appendix E: Entropy Documentation and Assessment

## E.1. Design Description

Documentation shall include the design of each entropy source as a whole, including the interaction of all entropy source components. Any information that can be shared regarding the design should also be included for any third-party entropy sources that are included in the product.

The documentation will describe the operation of the entropy source to include 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 entropy comes from, where the entropy output is passed next, any post-processing of the raw outputs (hash, XOR, etc.), if/where it is stored, and finally, how it is output from the entropy source. Any conditions placed on the process (e.g., blocking) should also be described in the entropy source design. Diagrams and examples are encouraged.

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

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

There should be a technical argument for where the unpredictability in the source comes from and why there is confidence in the entropy source delivering sufficient entropy for the uses made of the RBG output (by this particular TOE). This argument will include a description of the expected min-entropy rate (i.e. the minimum entropy (in bits) per bit or byte of source data) and explain 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.

The amount of information necessary to justify the expected min-entropy rate depends on the type of entropy source included in the product.

For developer-provided entropy sources, in order to justify the min-entropy rate, it is expected that a large number of raw source bits will be collected, statistical tests will be performed, and the min-entropy rate determined from the statistical tests. While no particular statistical tests are required

at this time, it is expected that some testing is necessary in order to determine the amount of min-entropy in each output.

For third-party provided entropy sources, in which the TOE vendor has limited access to the design and raw entropy data of the source, the documentation will indicate an estimate of the amount of min-entropy obtained from this third-party source. It is acceptable for the vendor to "assume" an amount of min-entropy, however, this assumption must be clearly stated in the documentation provided. In particular, the min-entropy estimate must be specified and the assumption included in the ST.

Regardless of the type of entropy source, the justification will also include how the DRBG is initialized with the entropy stated in the ST, for example by verifying that the min-entropy rate is multiplied by the amount of source data used to seed the DRBG or that the rate of entropy expected based on the amount of source data is explicitly stated and compared to the statistical rate. If the amount of source data used to seed the DRBG is not clear or the calculated rate is not explicitly related to the seed, the documentation will not be considered complete.

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

The entropy rate may be affected by conditions outside the control of the entropy source itself. For example, voltage, frequency, temperature, and elapsed time after power-on are just a few of the factors that may affect the operation of the entropy source. As such, 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

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 start up, 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 Document

## F.1. Key Management Description

The description will provide the following information for all keys in the key chain:

- The purpose of the key
- If the key is stored in non-volatile 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
- How and when the key may be shared
- Key destruction description

The description will also describe the following topics:

- A description of all authorization factors that are supported by the product and how each factor is handled, including any conditioning and combining performed.
- If validation is implemented, 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.
- The authorization process that leads to the decryption of the BEV or DEK. This section shall detail the key chain used by the product. It shall describe which keys are used in the protection of the BEV or DEK and how they meet the encryption or derivation requirements including the direct chain from the initial authorization to the BEV or DEK. 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.
- The diagram and essay will clearly illustrate the key hierarchy to ensure that at no point the chain could be broken without a cryptographic exhaust or all of the initial authorization values and the effective strength of the BEV or DEK is maintained throughout the key chain.
- A description of the data encryption engine, its components, and details about its implementation (e.g. initialization of the product, drivers, libraries (if applicable), logical interfaces for encryption/decryption, and how resources to be encrypted are identified). 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. The description should be detailed enough to verify all platforms ensure that when the user enables encryption, the product encrypts all selected resources.
- 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 non-volatile memory.

# F.2. Key Management Diagram:

- The diagram will include all keys from the initial authorization factor(s) to the BEV or DEK 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 options from key chaining requirement. The diagram should indicate the input used to derive or decrypt each key in the chain.
- A functional (block) diagram showing the main components (such as memories and processors) the initial steps needed for the activities the TOE performs to ensure it encrypts the targeted resources when a user or administrator first provisions the product.

# Appendix G: Glossary

For the purpose of this cPP, the following terms and definitions given in *some specific references* apply. If the same terms and definitions are given in those references, terms and definitions that fit the context of this cPP take precedence.

**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.

**Border Encryption Value**

A secret value passed to a storage encryption component such as a self-encrypting storage device [CPP_FDE_EE_V2.0]

**cPP**

collaborative Protection Profile

**CBC**

Cipher Block Chaining

**CEM**

Common Methodology for Information Security Evaluation

**CNSSP**

Committee on National Security Systems Policy

**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 1.4.2, “USE CASE 2: Conditionally Mandatory Use Cases” which, if present in the TOE, must be included in its evaluated configuration.

**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.

**DEK**

Data Encryption Key

**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.

**Document Processing Job**

A User request to the TOE to perform a Document Processing operation on a Document.

**DSA**

Digital Signature Algorithm

**ECDSA**

Elliptic Curve Digital Signature Algorithm

**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”

**FEK**

File Encryption Key

**FIPS**

Federal Information Processing Standards

**GCM**

Galois/Counter Mode

**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]

**HMAC**

Keyed-hash message authentication code

**HTTPS**

Hypertext Transfer Protocol Secure

**Intermediate Key**

A key used in a point between the initial user authorization and the DEK. [CPP_FDE_EE_V2.0]

**Internal Authentication**

Identification and authentication function that is wholly contained within the TOE.

**IPsec**

IP security

**iTC**

international Technical Community

**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.

**KDF**

Key Derivation Functions

**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]

**NAT**

Network address translation

**Network Printing**

Printing operation that has been initiated by a Network User.

**Network User**

A User who interacts with the HCD over a network.

**NIST**

National Institute of Standards and Technology

**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.

**NTP**

Network Time Protocol

**Operational Environment**

Environment in which the TOE is operated. [CC]

**Optional Use**

One of the uses described in Section 1.4.3, “USE CASE 3: Optional Use Cases” which may be present in the TOE, and may optionally be included in its evaluated configuration.

**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 1.4.1, “USE CASE 1: Required Use Cases” which must be present in the TOE in its evaluated configuration.

**RFC**

Request for Comments

**RNG**

Random Number Generator

**Root of Trust**

Highly reliable hardware, firmware, and software components that perform specific, critical security functions. Because roots of trust are inherently trusted, they must be secure by design. Roots of trust provide a firm foundation from which to build security and trust.

**RSA**

Rivest–Shamir–Adleman

**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]

**Security Target**

Implementation-dependent statement of security needs for a specific identified TOE. [CC]

**SED**

Self Encrypting Drive

**Servicing**

Performing repairs or preventative maintenance on the HCD.

**SPD**

Security Problem Definition

**SSH**

Secure Shell

**ST**

Security Target

**Standard Protection Profile**

A Protection Profile that is developed according to processes defined by NIAP.

**Submask**

A submask is a bit string that can be generated and stored in a numbers of ways, such as passphrases, tokens, etc. [CPP_FDE_EE_V2.0]

**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]

**TLS**

Transport Layer Security

**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]

**TPM**

Trusted Platform Module

**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]

**TSS**

TOE Summary Specification

**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]

**XTS**

XEX-based tweaked-codebook mode with ciphertext stealing

**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"

[CPP_FDE_EE_V2.0] collaborative Protection Profile for Full Drive Encryption – Encryption Engine, Version 2.0, September 09, 2016

[FAR] United States Federal Acquisition Regulations

# Appendix H: Acronyms

*Table 9. Acronyms*

| Acronym | Meaning |
|---|---|
| AES | Advanced Encryption Standard |
| ITSEF | IT Security Evaluation Facility |
| BEV | Border Encryption Value |
| CBC | Cipher Block Chaining |
| CC | Common Criteria |
| CCEVS | Common Criteria Evaluation and Validation Service |
| CEM | Common Methodology for Information Security Evaluation |
| CNSSP | Committee on National Security Systems Policy |
| COTS | Commercial Off-The-Shelf |
| cPP | collaborative Protection Profile |
| DEK | Data Encryption Key |
| DSA | Digital Signature Algorithm |
| EAL | Evaluation Assurance Level |
| ECDSA | Elliptic Curve Digital Signature Algorithm |
| FEK | File Encryption Key |
| FIPS | Federal Information Processing Standards |
| GCM | Galois/Counter Mode |
| HCD | Hardcopy Device |
| HMAC | keyed-hash message authentication code |
| HTTPS | Hypertext Transfer Protocol Secure |
| IPA | Information-technology Promotion Agency |
| I&A | Identification and Authentication |
| IPsec | IP security |
| IT | Information Technology |
| iTC | international Technical Community |
| JISEC | Japan Information technology Security Evaluation and Certification scheme |
| KDF | Key Derivation Function |
| KMD | Key Management Description |
| LAN | Local Area Network |

| Acronym | Meaning |
|---|---|
| LDAP | Lightweight Directory Access Protocol |
| MFD | Multifunction Device |
| MFP | Multifunction Printer, Multifunction Peripheral |
| NAT | Network address translation |
| NIAP | National Information Assurance Partnership |
| NIST | National Institute of Standards and Technology |
| NTP | Network Time Protocol |
| OCSP | Online Certificate Status Protocol |
| OSP | Organizational Security Policy |
| PP | Protection Profile |
| PSTN | Public Switched Telephone Network |
| RBG | Random Bit Generator |
| RFC | Request for Comments |
| RNG | Random Number Generator |
| RSA | Rivest–Shamir–Adleman |
| SAR | Security Assurance Requirement |
| SED | Self Encrypting Drive |
| SFP | Security Functional Policy |
| SFR | Security Functional Requirement |
| SPD | Security Problem Definition |
| SPP | Standard Protection Profile |
| SSH | Secure Shell |
| ST | Security Target |
| TLS | Transport Layer Security |
| TOE | Target of Evaluation |
| TPM | Trusted Platform Module |
| TSF | TOE Security Functionality |
| TSFI | TSF Interface |
| TSS | TOE Summary Specification |
| XTS | XEX-based tweaked-codebook mode with ciphertext stealing |

# Appendix I: Definitions and Rationale Tables

# I.1. User Definitions

There are two categories of Users defined in this cPP:

*Table 10. 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. A Normal User can be a Local User or a Network User as described in Section Section 1.3.2, "Operational Environment" |
| **U.ADMIN** | Administrator | A User who has been identified and authenticated and has an administrative role |

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.

# I.2. Asset Definitions

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

*Table 11. 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 |

A conforming TOE may define additional Asset categories.

## I.2.1. User Data

User Data are composed of two types:

*Table 12. 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 |

A conforming TOE may define additional types of User Data.

### I.2.2. TSF Data

TSF Data are composed of two types:

*Table 13. 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 |

A conforming TOE may define additional types of TSF Data, examples include:

*Table 14. Examples of TSF Data Categorization*

| Examples of TSF Protected Data | Examples of TSF Confidential Data |
|---|---|
| User and Administrator identification data | User and Administrator authentication data |
| Scan/fax/e-mail destination lists or address books | Credentials for accessing external devices (e.g., e-mail or file servers) |
| Job status logs | Job details and audit logs |
| Status of pending or stored jobs and documents | Access control lists |
| Device and network status information and configuration settings | Device and network management (e.g., Simple Network Management Protocol) authentication data |
| Device security status | Cryptographic keys |
| Device firmware and software | |

## I.3. Threat Definitions

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 15. 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 or the physical Nonvolatile Storage component. |
| **T.TSF_COMPROMISE** | An attacker may gain Unauthorized Access to TSF Data in the TOE through one of the TOE's interfaces or the physical Nonvolatile Storage component. |

| Designation | Definition |
|---|---|
| **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 firmware/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. |
| **T.WEAK_CRYPTO** | An attacker may exploit poorly chosen cryptographic algorithms, random bit generators, ciphers or key sizes. |

# I.4. Organizational Security Policy Definitions

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 16. 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 stored within the TOE as well as 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** | If the TOE stores User Document Data or Confidential TSF Data on Nonvolatile Storage Devices, it will encrypt such data on those devices. |
| **P.KEY_MATERIAL** | Cleartext keys, submasks, random numbers, or any other values that contribute to the creation of encryption keys for 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 Nonvolatile Storage Devices. |
| **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. |

| Designation | Definition |
|---|---|
| **P.ROT_INTEGRITY** | The vendor provides a Root of Trust (RoT) that is comprised of the TOE firmware, hardware, and pre-installed public keys or required critical security parameters. |

# I.5. Assumption Definitions

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

*Table 17. 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. |

# I.6. Definitions of Security Objectives for the TOE

*Table 18. 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 firmware/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. |

| Designation | Definition |
|---|---|
| **O.AUDIT** | The TOE shall generate audit data and store it internally as well as be capable of sending it to a trusted External IT Entity. |
| **O.STORAGE_ENCRYPTION** | If the TOE stores User Document Data or Confidential TSF Data in Nonvolatile Storage devices, then the TOE shall encrypt such data on those devices. |
| **O.KEY_MATERIAL** | 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 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 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. |
| **O.AUTH_FAILURES** (conditionally mandatory) | The TOE resists repeated attempts to guess authorization data by responding to consecutive failed attempts in a way that prevents an attacker from exploring a significant amount of the space of possible authorization data values. |
| **O.FW_INTEGRITY** | The TOE ensures its own integrity has remained intact and attests its integrity to outside parties on request. |
| **O.STRONG_CRYPTO** | The TOE implements strong cryptographic mechanisms and algorithms according to recognized standards, including support for random bit generation based on recognized standards and a source of sufficient entropy. The TOE uses key sizes that are recognized as providing sufficient resistance to current attack capabilities. |

# I.7. Definitions of Security Objectives for the Operational Environment

*Table 19. 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. |

| Designation | Definition |
|---|---|
| **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. |

# I.8. Security Objectives Tables

*Table 20. 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. |
| **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.WEAK_CRYPTO** An attacker may exploit poorly chosen cryptographic algorithms, random bit generators, ciphers or key sizes. | **O.STRONG_CRYPTO** implements strong cryptographic mechanisms to provide sufficient resistance to current attack capabilities. |
| **T.UNAUTHORIZED_UPDATE** An attacker may cause the installation of unauthorized firmware/software on the TOE. | **O.UPDATE_VERIFICATION** verifies the authenticity of firmware/software updates. |

| Threat/Policy/Assumption | Rationale |
| --- | --- |
| **T.NET_COMPROMISE**<br>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**<br>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.<br>**O.USER_I&A** provides the basis for authorization.<br>**O.ADMIN_ROLES** restricts the ability to authorize Users to authorized Administrators. |
| **P.AUDIT**<br>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.<br>**O.ACCESS_CONTROL** restricts access to audit data in the TOE to authorized Users.<br>**O.USER_AUTHORIZATION** provides the basis for authorization. |
| **P.COMMS_PROTECTION**<br>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**<br>If the TOE stores User Document Data or Confidential TSF Data on Nonvolatile Storage Devices, it will encrypt such data on those devices. | **O.STORAGE_ENCRYPTION** protects User Document Data and Confidential TSF Data stored in Nonvolatile Storage Devices from exposure if a device has been removed from the TOE and its Operational Environment. |
| **P.KEY_MATERIAL**<br>Cleartext keys, submasks, random numbers, or any other values that contribute to the creation of encryption keys for 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. |

| Threat/Policy/Assumption | Rationale |
|---|---|
| **P.FAX_FLOW** (conditionally mandatory)<br>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. |
| **P.IMAGE_OVERWRITE** (optional)<br>Upon completion or cancellation of a Document Processing job, the TOE shall overwrite residual image data from its Nonvolatile Storage Device. | **O.IMAGE_OVERWRITE** overwrites residual image data from Nonvolatile Storage Devices after Document Processing jobs are completed or cancelled. |
| **P.PURGE_DATA** (optional)<br>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. |
| **P.ROT_INTEGRITY**<br>The vendor provides a Root of Trust (RoT) that is comprised of the TOE firmware, hardware, and pre-installed public keys or required critical security parameters. | **[TBD]** |
| **A.PHYSICAL**<br>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**<br>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. |

| Threat/Policy/Assumption | Rationale |
| --- | --- |
| **A.TRUSTED_ADMIN**<br>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. |
| **A.TRAINED_USERS**<br>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.<br>**OE.USER_TRAINING** establishes responsibility of the TOE Owner to provide appropriate training for Users. |