Aruba Mobility Master with ArubaOS 8.2

Security Target

Version 1.3

July 2020

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Author	Description
1.0	4 Mar 2020	L Turner	Release for certification.
1.1	17 Apr 2020	L Turner	Certification updates.
1.2	4 May 2020	L Turner	Certification updates.
1.3	10 July 2020	L Turner	NIAP updates.

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the Aruba Mobility Master with ArubaOS 8.2 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 The Aruba Mobility Master simplifies the management of multiple Aruba controllers running ArubaOS 8 or later. Key features include a centralized dashboard to easily see and manage controllers deployed in multiple sites, a hierarchical configuration tool to pre-stage network deployments, and the ability to perform live firmware and feature upgrades during active user sessions. The addition of licensing pools simplifies the transfer of licenses between different controllers to quickly address expanded deployment needs.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	Aruba Mobility Master with ArubaOS 8.2 Build: 8.2.2.7
Security Target	Aruba Mobility Master with ArubaOS 8.2 Security Target, v1.3

1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) CC version 3.1 revision 5

b) CC Part 2 extended

c) CC Part 3 conformant

d) collaborative Protection Profile for Network Devices, v2.1

e) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name	Notes
TD0395	NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2	FCS_TLSS_EXT.2 not claimed.
TD0396	NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2	TLSC not claimed.
TD0397	NIT Technical Decision for Fixing AES-CTR Mode Tests	
TD0398	NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	
TD0399	NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)	

TD #	Name	Notes
TD0400	NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment	
TD0401	NIT Technical Decision for Reliance on external servers to meet SFRs	
TD0402	NIT Technical Decision for RSA-based FCS_CKM.2 Selection	
TD0407	NIT Technical Decision for handling Certification of Cloud Deployments	TOE is not a cloud deployment.
TD0408	NIT Technical Decision for local vs. remote administrator accounts	
TD0409	NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication	
TD0410	NIT technical decision for Redundant assurance activities associated with FAU_GEN.1	
TD0411	NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 - Server and client side seem to be confused	SSHC not claimed.
TD0412	NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy	
TD0423	NIT Technical Decision for Clarification about application of Rfl#201726rev2	
TD0424	NIT Technical Decision for NDcPP v2.1 Clarification - FCS_SSHC/S_EXT1.5	
TD0425	NIT Technical Decision for Cut-and-paste Error for Guidance AA	
TD0447	NIT Technical Decision for Using 'diffie-hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7	
TD0450	NIT Technical Decision for RSA-based ciphers and the Server Key Exchange message	
TD0451	NIT Technical Decision for ITT Comm UUID Reference Identifier	
TD0453	NIT Technical Decision for Clarify authentication methods SSH clients can use to authenticate SSH se	SSHC not claimed.
TD0475	NIT Technical Decision for Separate traffic consideration for SSH rekey	

TD #	Name	Notes
TD0477	NIT Technical Decision for Clarifying FPT_TUD_EXT.1 Trusted Update	
TD0478	NIT Technical Decision for Application Notes for FIA_X509_EXT.1 iterations	
TD0480	NIT Technical Decision for Granularity of audit events	
TD0481	NIT Technical Decision for FCS_(D)TLSC_EXT.X.2 IP addresses in reference identifiers	TLSC not claimed.
TD0482	NIT Technical Decision for Identification of usage of cryptographic schemes	
TD0483	NIT Technical Decision for Applicability of FPT_APW_EXT.1	
TD0484	NIT Technical Decision for Interactive sessions in FTA_SSL_EXT.1 & FTA_SSL.3	

1.4 Terminology

Table 3: Terminology

Term	Definition
AP	Access Point (wireless)
Aruba AP	Aruba APs provide wireless connectivity to wireless clients.
ArubaOS	The ArubaOS is a suite of applications that runs on Mobility Master and Mobility Controllers that allows administrators to configure and manage the wireless and mobile user environment.
Aruba Mobility Controller	Aruba Mobility Controllers serve as a gateway between wired and wireless networks and provides command and control over Aruba APs within an Aruba dependent wireless network.
Aruba Mobility Master	The Aruba Mobility Master allows for centralized management of multiple Aruba controllers.
CC	Common Criteria
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
NDcPP	collaborative Protection Profile for Network Devices
PP	Protection Profile

Term	Definition
TOE	Target of Evaluation
TSF	TOE Security Functionality
WLAN	Wireless Local Area Network

2 TOE Description

2.1 Type

4 The TOE is a network device that provides centralized management of multiple Aruba Mobility Controllers.

2.2 Usage

5 The TOE is deployed within a network that provides connectivity to the managed Aruba controllers. The TOE's web-based GUI is the primary user interface, as shown in Figure 1.

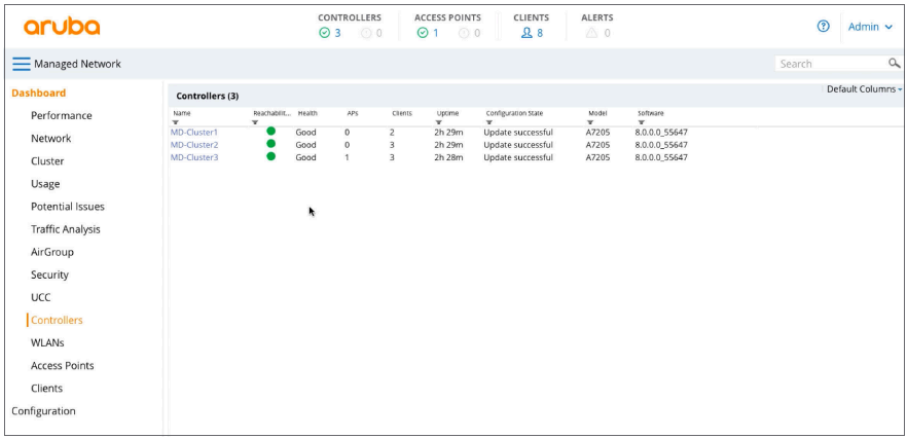


Figure 1: TOE User Interface

6 The TOE interfaces within the scope of evaluation are shown in Figure 2.

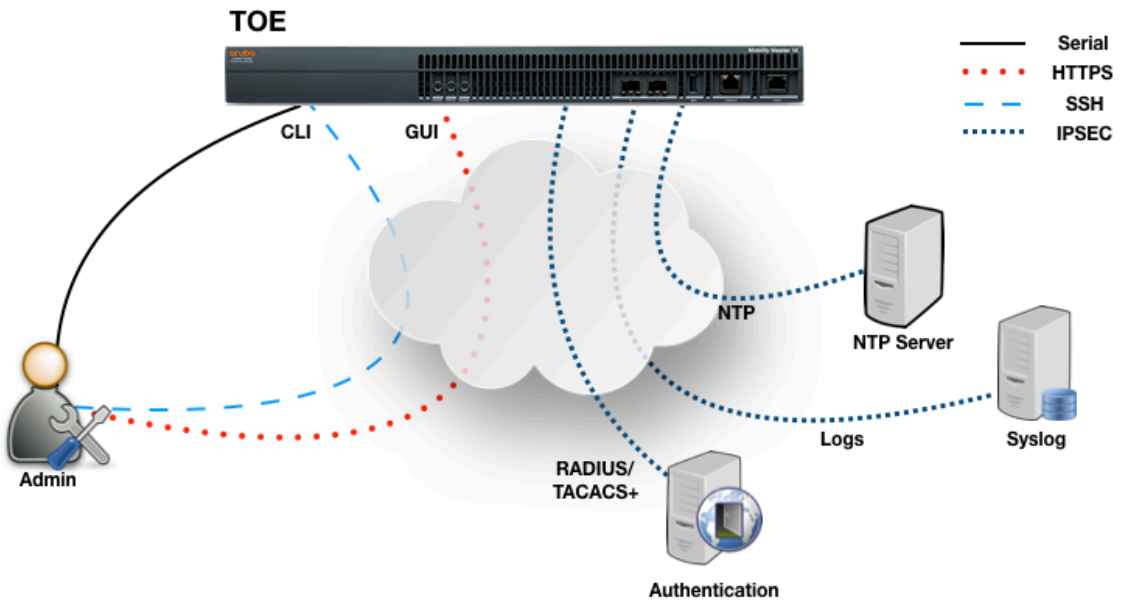


Figure 2: TOE interfaces

7 The TOE interfaces are as follows:

a) **CLI.** Administrative CLI via direct serial connection or SSH.

b) **GUI.** Administrative web GUI via HTTPS / TLS.

c) **RADIUS/TACACS+.** Authentication with a remote server via IPsec.

d) **Logs.** Logs sent to syslog via IPsec.

e) **NTP.** Time synchronization with an NTP server via IPsec.

2.3 Security Functions

8 The TOE provides the following security functions:

a) **Protected Communications.** The TOE protects the integrity and confidentiality of communications as noted in section 2.2 above.

b) **Secure Administration.** The TOE enables secure management of its security functions, including:

i) Administrator authentication with passwords

ii) Configurable password policies

iii) Role Based Access Control

iv) Access banners

v) Management of critical security functions and data

vi) Protection of cryptographic keys and passwords

c) **Trusted Update.** The TOE ensures the authenticity and integrity of software updates through digital signatures.

d) **System Monitoring.** The TOE generates logs of security relevant events. The TOE stores logs locally and is capable of sending log events to a remote audit server.

e) **Self-Test.** The TOE performs a suite of self-tests to ensure the correct operation and enforcement of its security functions.

f) **Cryptographic Operations.** The TOE implements a cryptographic module. Relevant Cryptographic Algorithm Validation Program (CAVP) certificates are shown in Table 4 and Table 5.

Table 4: CAVP Certificates

Module Name	Certificates
ArubaOS OpenSSL Module	C1975
ArubaOS Crypto Module	C1976

2.4 Physical Scope

9 The physical boundary of the TOE includes the appliance models shown in Table 5 executing ArubaOS 8.2. The TOE is shipped to uses via commercial courier.

10 The ArubaOS consists of a base software package with add-on software modules that can be activated by installing the appropriate licenses. The following modules are required to be licensed and installed in the CC evaluated configuration:

a) **Advanced Cryptography.** Required for SuiteB, AES-GCM and ECDSA functionality.

Table 5: TOE models

Model	Platform	Notes on Differences	CAVP
MM-HW-1K-F1	Intel Xeon E5-2609v4	Number of supported devices, clients and controllers.	C1975 C1976
MM-HW-5K-F1	Intel Xeon E5-2620v4		
MM-HW-10K-F1	Intel Xeon E5-2650v4		

2.4.1 Guidance Documents

11 The TOE includes the following guidance documents (PDF):

a) ArubaOS 8.2.0.x Command Line Interface Reference Guide, Revision 09

b) ArubaOS 8.2.0.x Getting Started Guide, Revision 02

c) ArubaOS 8.2.0.x User Guide, Revision 10

d) ArubaOS 8.x Syslog Message Guide, Revision 01

e) Mobility Master Hardware Appliance Installation Guide, Revision 02

f) Aruba OS 8.2 Supplemental Guidance - Common Criteria Configuration Guidance, v1.8

2.4.2 Non-TOE Components

12 The TOE operates with the following components in the environment:

a) **Audit Server.** The TOE is capable of sending audit events to a Syslog server.

b) **NTP Server.** The TOE synchronizes time with an NTP server.

c) **Authentication Server.** The TOE is able to utilize RADIUS and TACACs+ servers to authenticate users.

d) **Mobility Controllers.** The TOE manages Aruba controllers running ArubaOS 8 or later. Note: The TOE security functions are not reliant on the presence of Mobility Controllers.

2.5 Logical Scope

13 The logical scope of the TOE comprises the security functions defined in section 2.3.

2.5.1 Functions not included in the TOE Evaluation

14 The TOE is capable of a variety of functions which are not covered by the NDcPP. Only the security functions defined in section 2.3 have been examined as part of this evaluation.

15 IKEv1 must not be used in the evaluated configuration.

3 Security Problem Definition

16 The Security Problem Definition is reproduced from section 4 of the NDcPP.

3.1 Threats

Table 6: Threats

Identifier	Description
T.UNAUTHORIZED_ ADMINISTRATOR_ ACCESS	Threat agents may attempt to gain Administrator access to the network device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between network devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_ CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_ COMMUNICATION_ CHANNELS	Threat agents may attempt to target network devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the network device itself.
T.WEAK_ AUTHENTICATION_ ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.
T.UPDATE_ COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the network device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and

Identifier	Description
	the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_ FUNCTIONALITY_ COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_ CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_ FUNCTIONALITY_ FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

3.2 Assumptions

Table 7: Assumptions

Identifier	Description
A.PHYSICAL_ PROTECTION	The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.
A.LIMITED_ FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_ TRAFFIC_ PROTECTION	A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g., firewall).

Identifier	Description
A.TRUSTED_ ADMINISTRATOR	The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.
A.REGULAR_ UPDATES	The network device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_ CREDENTIALS_ SECURE	The Administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.
A.RESIDUAL_ INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

3.3 Organizational Security Policies

Table 8: Organizational Security Policies

Identifier	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

4 Security Objectives

17 The security objectives are reproduced from section 5 of the NDcPP.

Table 9: Security Objectives for the Operational Environment

Identifier	Description
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_ PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

Identifier	Description
OE.NO_THRU_ TRAFFIC_ PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDEN TIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESIDUAL_INFO RMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

5 Security Requirements

5.1 Conventions

18 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

e) **Iteration.** Indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

19 **Note:** Selection and assignment operations performed within the Security Target are denoted within brackets []. Operations shown without brackets are reproduced from the NDcPP.

5.2 Extended Components Definition

20 Refer to Annex A: Extended Components Definition.

5.3 Functional Requirements

Table 10: Summary of SFRs

Requirement	Title
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES 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_HTTPS_EXT.1	HTTPS Protocol
FCS_IPSEC_EXT.1	IPsec Protocol

Requirement	Title
FCS_NTP_EXT.1	NTP Protocol
FCS_RBG_EXT.1	Random Bit Generation
FCS_SSHS_EXT.1	SSH Server Protocol
FCS_TLSS_EXT.1	TLS Server Protocol
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1/ManualUpdate	Management of security functions behaviour
FMT_MOF.1/Functions	Management of security functions behaviour
FMT_MTD.1/CoreData	Management of TSF Data
FMT_MTD.1/CryptoKeys	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.2	Restrictions on Security Roles
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF testing
FPT_TUD_EXT.1	Extended: Trusted update
FPT_STM_EXT.1	Reliable Time Stamps
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination

Requirement	Title
FTA_SSL.4	User-initiated Termination
FTA_TAB.1	Default TOE Access Banners
FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1/Admin	Trusted Path

5.3.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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;

c) All administrative actions comprising:

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
- *Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- *Resetting passwords (name of related user account shall be logged).*
- no other actions;

d) *Specifically defined auditable events listed in ~~**Table 2**~~ **Table 11**.*

Table 11: Audit Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_NTP_EXT.1	Configuration of a new time server Removal of configured time server	Identity if new/removed time server
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Functions	Modification of the behaviour of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.

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, 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 cPP/ST, *information specified in column three of* ***~~Table 2~~ Table 11***.

FAU_GEN.2 User Identity Association

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.

FAU_STG_EXT.1 Protected Audit Event Storage

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.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself.[

- TOE shall consist of a single standalone component that stores audit data locally]

FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit records according to the following rule: [*overwrite oldest record first*], [*no other action*]] when the local storage space for audit data is full.

5.3.2 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

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

- 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” [P-256, P-384] 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 Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3

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

FCS_CKM.2 Cryptographic Key Establishment

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

- 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 2.1
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;
- Finite field-based key establishment schemes that meet the following: , “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;
- Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3;

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

FCS_CKM.4 Cryptographic Key Destruction

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

- *For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of [zeroes]];*
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [*
    - *logically addresses the storage location of the key and performs a [single overwrite consisting of [zeroes]];*

] that meets the following: *No Standard*.

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

FCS_COP.1.1/DataEncryption The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in [CBC, CTR, GCM] mode* and cryptographic key sizes *[128 bits, 256 bits]* that meet the following: *AES as specified in ISO 18033-3, [CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772*].

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

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

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [*2048 bits or greater*],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits or greater*]

] that meet the following: [

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

FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

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

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

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

FCS_HTTPS_EXT.1 HTTPS 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.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [not establish the connection] if the peer certificate is deemed invalid.

FCS_IPSEC_EXT.1 IPsec Protocol

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 [tunnel mode].

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [AES-CBC-128, AES-CBC-256 (specified in RFC 3602)] together with a Secure Hash Algorithm (SHA)-based HMAC [HMAC-SHA-1] and [AES-GCM-128, AES-GCM-256 (specified in RFC 4106)].

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [

- IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [RFC 4868 for hash functions]

].

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [IKEv2] protocol uses the cryptographic algorithms [AES-CBC-128, AES-CBC-256 (specified in RFC 3602].

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [

- IKEv2 SA lifetimes can be configured by a Security Administrator based on [
    - length of time, where the time values can be configured within [*1-24*] hours

    ]

].

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [

- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [
    - number of bytes;
    - length of time, where the time values can be configured within [*1-8*] hours;

    ]

].

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange (“x” in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [*160, 256, 384*] bits.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in [IKEv2] exchanges of length [

- 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) [14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP)].

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 [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 [IKEv2 CHILD_SA] connection.

FCS_IPSEC_EXT.1.13 The TSF shall ensure that all IKE protocols perform peer authentication using [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [Pre-shared Keys].

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: [Distinguished Name (DN)] and [no other reference identifier type].

FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s) [NTP v4 (RFC 5905)].

FCS_NTP_EXT.1.2 The TSF shall update its system time using [

- [IPsec] to provide trusted communication between itself and an NTP time source.

].

FCS_NTP_EXT.1.3 The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4 The TSF shall support configuration of at least three (3) NTP time sources.

FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [CTR_DRBG (AES)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[2] hardware based noise sources] with a minimum of [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.

FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) [4251, 4252, 4253, 4254, 4344, 5647, 5656].

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, [password based].

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*256k*] 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: [aes128-cbc, aes256-cbc, aes128-ctr, aes256-ctr].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa] 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 [hmac-sha1, hmac-sha1-96] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1] and [diffie-hellman-group14-sha256] 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 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: The above SFR is altered by TD0398, TD0424 and TD0475.

FCS_TLSS_EXT.1 TLS Server Protocol with mutual authentication

FCS_TLSS_EXT.1.1 The TSF shall implement [TLS 1.2 (RFC 5246)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:[

- 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_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_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].

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [TLS 1.1].

FCS_TLSS_EXT.1.3 The TSF shall [perform RSA key establishment with key size [2048 bits]; generate EC Diffie-Hellman parameters over NIST curves [secp256r1 and no other curves]].

5.3.3 Identification and Authentication (FIA)

FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1-10*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].

Application Note: The above SFR is altered by TD0408.

FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”];

b) Minimum password length shall be configurable to between [*8*] and [*32] characters.*

FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [[*no other action*]]

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local [password-based] authentication mechanism to perform local administrative user authentication.

Application Note: The above SFR is altered by TD0408.

FIA_UAU.7 Protected Authentication Feedback

FIA_UAU.7.1 The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress **at the local console**.

FIA_X509_EXT.1/Rev X.509 Certificate Validation

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 presence of the basicConstraints extension and that the CA flag is set to TRUE.
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 6960]
- 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.

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [HTTPS, TLS, IPsec], and [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 [not accept the certificate].

FIA_X509_EXT.3 X.509 Certificate Requests

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

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

5.3.4 Security Management (FMT)

FMT_MOF.1/ManualUpdate Management of security functions behaviour

FMT_MOF.1.1/ManualUpdate The TSF shall restrict the ability to enable the functions *to perform manual updates* to *Security Administrators*.

FMT_MOF.1/Functions Management of security functions behaviour

FMT_MOF.1.1/Functions The TSF shall restrict the ability to [modify the behaviour of] the functions [transmission of audit data to an external IT entity] to *Security Administrators*.

FMT_MTD.1/CoreData Management of TSF Data

FMT_MTD.1.1/CoreData The TSF shall restrict the ability to *manage* the *TSF data* to *Security Administrators*.

FMT_MTD.1/CryptoKeys Management of TSF data

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to *manage* the *cryptographic keys to Security Administrators*.

FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [
  - Ability to configure audit behaviour;
  - Ability to manage the cryptographic keys;
  - Ability to set the time which is used for time-stamps;
  - Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
  - Ability to import X.509v3 certificates to the TOE's trust store;]

FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator*.

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely*

are satisfied.

5.3.5 Protection of the TSF (FPT)

FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

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

FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1 The TSF shall store administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of administrative plaintext passwords.

FPT_TST_EXT.1 TSF testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [during initial start-up (on power on)] to demonstrate the correct operation of the TSF: [

- *Firmware integrity tests*
- *Cryptographic module tests*
- *BIOS tests*].

FPT_TUD_EXT.1 Trusted update

FPT_TUD_EXT.1.1 The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [no other TOE firmware/software version].

FPT_TUD_EXT.1.2 The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [no other update mechanism].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature mechanism] prior to installing those updates.

FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1.1 The TSF shall be able to provide reliable time stamps for its own use.

FPT_STM_EXT.1.2 The TSF shall [synchronise time with an NTP server].

5.3.6 TOE Access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [

- terminate the session]

after a Security Administrator-specified time period of inactivity.

FTA_SSL.3 TSF-initiated Termination

FTA_SSL.3.1 The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity*.

FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1 Refinement: The TSF shall allow **Administrator**-initiated termination of the **Administrator's** own interactive session.

FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1 Before establishing **an administrative user** session the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

5.3.7 Trusted path/channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall **be capable of using [IPsec] to provide** a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [authentication server [*Aruba Mobility Controllers]*]** 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 The TSF shall permit **the TSF or the authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*audit, authentication, management of mobility controllers*].

FTP_TRP.1 /Admin Trusted Path

FTP_TRP.1.1/Admin The TSF shall **be capable of using [SSH, HTTPS] to** provide a communication path between itself and **authorized** 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 provides detection of modification of the channel data**.

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

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

5.4 Assurance Requirements

21 The TOE security assurance requirements are summarized in Table 12.

Table 12: Assurance Requirements

Assurance Class	Components	Description
Security Target Evaluation	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.1	Security Objectives for the operational environment
	ASE_REQ.1	Stated Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

22 In accordance with section 7.1 of the NDcPP, the following refinement is made to ASE:

a) **ASE_TSS.1.1C Refinement:** The TOE summary specification shall describe how the TOE meets each SFR. **In the case of entropy analysis, the TSS is used in conjunction with required supplementary information on Entropy.**

6 TOE Summary Specification

23 The following describes how the TOE fulfils each SFR included in section 5.3.

6.1 Security Audit

6.1.1 FAU_GEN.1

24 The TOE generates the audit records specified at Table 11.

25 The following information is logged as a result of the Security Administrator generating/importing or deleting cryptographic keys:

a) **Generate CSR.** Action and key reference.

b) **Import Certificate.** Action and key reference.

c) **Import CA Certificate.** Action and key reference.

6.1.2 FAU_GEN.2

26 The TOE includes the user identity in audit events resulting from actions of identified users.

6.1.3 FAU_STG_EXT.1

27 The Security Administrator can configure the TOE to send logs to a Syslog server. Log events are sent in real-time. Logs are sent via IPsec.

28 The TOE stores audit records locally and provides CLI and Web UI capabilities for the administrator to view the contents of the audit trail. For local storage, the maximum log file size for all processes is ARUBA_MAX_LOG_FILE_SIZE (i.e. 95,304 bytes). However, the security.log, system.log and user-debug logs have a maximum file size given by A_MAX_SECURITY_USER_DEBUG_LOG_FILE_SIZE (i.e. 1000KB).

29 When the local audit data store is full, the TOE will overwrite audit records starting with the oldest audit record.

30 Only authorized administrators may view audit records and no capability to modify the audit records is provided.

6.2 Cryptographic Support

6.2.1 FCS_CKM.1

31 The TOE supports key generation for the following asymmetric schemes:

a) **RSA 2048-bit.** Used in SSH and TLS.

b) **ECC P-256/P-384.** Used in TLS and IPsec.

c) **FFC 2048-bit.** Diffie-Hellman used in TLS and IPsec.

d) **Diffie-Hellman group 14.** Diffie-Hellman group 14 used in SSH and IPsec.

6.2.2 FCS_CKM.2

32 The TOE supports the following key establishment schemes:

a) **RSA schemes.** Used in TLS. TOE is sender.

b) **ECC schemes.** Used in TLS. TOE is sender.

c) **FFC schemes.** Diffie-Hellman used in TLS and IPsec. TOE is sender.

d) **Diffie-Hellman group 14.** Used in SSH and IPsec. The TOE meets RFC 3526 Section 3 by implementing the 2048-bit Modular Exponential (MODP) Group. TOE is sender.

33 Table 13 below identifies the scheme being used by each service.

Table 13: Key Agreement Mapping

Scheme	SFR	Service
RSA	FCS_TLSS_EXT.1	GUI / Administration
ECC	FCS_TLSS_EXT.1	GUI / Administration
FFC	FCS_TLSS_EXT.1	GUI / Administration
	FCS_IPSEC_EXT.1	Audit Server Authentication Server Aruba Mobility Controller
Diffie-Hellman group 14	FCS_SSHS_EXT.1	CLI / Administration
	FCS_IPSEC_EXT.1	Audit Server Authentication Server Aruba Mobility Controller

6.2.3 FCS_CKM.4

34 Table 15 shows the origin, storage location and destruction details for cryptographic keys.

6.2.4 FCS_COP.1/DataEncryption

35 The TOE provides symmetric encryption and decryption capabilities using 128 and 256 bit AES in CBC, CTR and GCM mode. AES is implemented in the following protocols: TLS, SSH and IPsec.

36 The relevant NIST CAVP certificate numbers are listed Table 4.

6.2.5 FCS_COP.1/SigGen

37 The TOE provides cryptographic signature generation and verification services using:

a) RSA Signature Algorithm with key size of 2048 bits

b) ECDSA Signature Algorithm with NIST curves P-256, P-384.

38 Signature verification services are used in the TLS, SSH and IPsec protocols. Additionally, RSA signature verification is used for trusted updates.

39 The relevant NIST CAVP certificate numbers are listed in Table 4.

6.2.6 FCS_COP.1/Hash

40 The TOE provides cryptographic hashing services using SHA-1, SHA-256 and SHA-384.

41 SHA is implemented in the following parts of the TSF:

a) TLS, SSH and IPsec; and

b) Digital signature verification as part of trusted update validation

42 The relevant NIST CAVP certificate numbers are listed in Table 4.

6.2.7 FCS_COP.1/KeyedHash

43 The TOE provides keyed-hashing message authentication services using HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384.

44 HMAC is implemented in the following protocols: TLS, IPsec and SSH.

45 The characteristics of the HMACs used in the TOE are given in Table 14.

Table 14: HMAC Characteristics

Algorithm	Block Size	Key Size	Digest Size
HMAC-SHA-1	512 bits	160 bits	160 bits
HMAC-SHA-256	512 bits	256 bits	256 bits
HMAC-SHA-384	1024 bits	384 bits	384 bits

46 The relevant NIST CAVP certificate numbers are listed in Table 4.

6.2.8 FCS_HTTPS_EXT.1

47 The TOE web GUI is accessed via an HTTPS connection. The TOE does not use HTTPS in a client capacity. The TOE's HTTPS protocol complies with RFC 2818.

48 RFC 2818 specifies HTTP over TLS. The majority of RFC 2818 is spent on discussing practices for validating endpoint identities and how connections must be setup and torn down. The TOE web GUI operates on an explicit port designed to natively speak TLS: it does not attempt STARTTLS or similar multi-protocol negotiation which is described in section 2.3 of RFC 2818. The web server attempts to send closure Alerts prior to closing a connection in accordance with section 2.2.2 of RFC 2818.

6.2.9 FCS_IPSEC_EXT.1

49 The TOE includes an implementation of IPsec in accordance with RFC 4301 for security. The TOE supports IPsec for tunnel mode. The IPsec ESP protocol is implemented in conjunction with AES-CBC-128 and AES-CBC-256 (as specified by RFC 3602) together with the following SHA-based HMAC algorithms: HMAC-SHA-1 (truncated) and with AES-GCM-128 and AES-GCM-256 (as specified by RFC 4106).

50 The TOE implements IKEv2, with support for NAT traversal, as defined in RFC 5996 and RFC 4868 for hash functions. Diffie-Hellman (DH) Groups 14, 19, and 20 are supported as are RSA and ECDSA certificates and pre-shared key IPsec authentication. The TOE uses the AES-CBC-128 and AES-CBC-256 algorithms as specified in RFC 3602 to encrypt the payload.

51 The TOE generates the secret value x used in the IKEv2 Diffie-Hellman key exchange ('x' in gx mod p) using the FIPS validated RBG specified in FCS_RBG_EXT.1 and having possible lengths of 112, 192 or 384 bits (for DH Groups 14, 19, and 20, respectively). The TOE generates nonces used in the IKEv2 exchanges of length 112 bits, 128 bits and 192 bits and at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash.

52 Lifetimes for IKEv2 SAs are established during configuration of the IKE policies via the CLI function by an authorized administrator and can be configured with 1-24 hours for the IKEv2 IKE_SA and within 1-8hrs for the IKEv2 IKE_CHILD SA. The TOE also supports volume-based rekeying for the IKEv2 IKE_CHILD SA. In the IKEv2 IKE_SA and IKE_CHILD exchanges, the TOE and peer will agree on the best DH group both can support. When the TOE initiates IKE negotiation, the DH group is sent in order according to the peer's configuration. When the TOE receives an IKE proposal, it will select the first match and the negotiation will fail if there is no match.

53 The TOE checks to ensure the negotiated symmetric algorithm in the IKEv2 CHILD_SA is less than or equal to the strength of the IKEv2 IKE_SA.

54 Pre-shared keys used for IPsec can be constructed of essentially any alphabetic character (upper and lower case), numerals, and special characters (e.g., "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")") and can be anywhere from 6-160 characters in length (e.g., 22 characters). The TOE requires suitable keys to be entered by an authorized administrator using a Web GUI or CLI function.

55 The TOE will only establish a trusted IPsec channel if the presented identifier in the received certificate matches the configure reference identifier, where the presented and reference identifiers are of the following type: Distinguished Name (DN). Fields within the DN are not individually selectable; the DN must be an exact match for the entire DN string.

56 The TOE implements an SPD to determine what traffic gets protected with IPsec, what gets bypassed, and what gets dropped. The SPD is achieved via the routing table and firewall policies. The TOE administrator implicitly configures the IPsec SPD via the routing table and firewall policies and includes a final rule that causes the network packet to be discarded if no other rules are matched.

6.2.10 FCS_NTP_EXT.1

57 The TOE supports NTP v4 by implementing the ntpd Linux daemon in client mode. The TOE supports IPsec between itself and the NTP server per FCS_IPSEC.1.

6.2.11 FCS_RBG_EXT.1

58 The TOE contains a CTR_DRBG that is seeded from two hardware entropy sources. Entropy from the noise source is used to seed the DRBG with 256 bits of full entropy.

59 Additional detail is provided the proprietary Entropy Description.

6.2.12 FCS_SSHS_EXT.1

60 The TOE implements SSH in compliance with RFCs 4251, 4252, 4253, 4254, 4344, 5647 and 5656.

61 The TOE supports password-based or public key (SSH-RSA) authentication.

62 The TOE examines the size of each received SSH packet. If the packet is greater than 256KB, it is automatically dropped.

63 The TOE utilises AES-CBC-128, AES-CBC-256, AES-128-CTR and AES-256-CTR for SSH encryption.

64 The TOE provides data integrity for SSH connections via HMAC-SHA1 and HMAC-SHA1-96.

65 The TOE supports diffie-hellman-group14-sha1 and diffie-hellman-group14-sha256 for SSH key exchanges.

66 The TOE will re-key SSH connections after 1 hour of after an aggregate of 1 gig of data has been exchanged (whichever occurs first).

6.2.13 FCS_TLSS_EXT.1

67 The TOE operates as a TLS server for the web GUI trusted path.

68 The server only allows TLS protocol version 1.2 (rejecting any other protocol version) and is restricted to the following ciphersuites by default:

a) TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

b) TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268

c) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

d) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492

e) TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246

f) TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246

g) TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246

h) TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246

i) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

j) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

k) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

l) TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

69 Ciphersuites are not user-configurable.

70 The TOE performs RSA key establishment with key size 2048 bits and generates DH parameters over NIST curves secp256r1.

6.3 Identification and Authentication

6.3.1 FIA_PMG_EXT.1

71 The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”.

72 The minimum password length is settable by the Administrator and can range from 6 to 128 characters

6.3.2 FIA_UIA_EXT.1

73 The TOE requires all users to be successfully identified and authenticated. The TOE warning banner may be viewed prior to authentication.

74 Administrative access to the TOE is facilitated through one of several interface:

a) Directly connecting to the TOE appliance

b) Remotely connecting to the TOE appliance via SSHv2

c) Remotely connecting to appliance GUI via HTTPS

75 In addition to the local authentication mechanism described by FIA_UAU_EXT.2, the TOE supports RADIUS and TACACS+ authentication servers. A trusted channel using IPsec is established between the TOE and the external authentication server.

76 Remote administrators are configured as users who have privileges to access the CLI and Web GUI administration interfaces and who are authenticated as users using the local database or an external authentication server. The remote administrators authenticate as users using a username and password via Web GUI and username/password or public key for SSH. Direct console to the CLI only supports username/password.

77 A successful logon takes place when a recognized username/password combination is provided and/or a recognized X.509 client certificate is presented by the administrator's web browser or SSH client.

6.3.3 FIA_UAU_EXT.2

78 Regardless of the interface at which the administrator interacts, the TOE prompts the user for a credential. Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. No TOE administrative access is permitted until an administrator is successfully identified and authenticated.

79 The TOE provides a local password-based authentication mechanism (in addition to remote authentication servers described at FIA_UIA_EXT.1).

80 The process for authentication is the same for administrative access whether administration is occurring via direct connection or remotely. At initial login, the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative credential associated with the user account (e.g. password or SSH public/private key response). The TOE then either grants administrative access (if the combination of username and credential is correct) or indicates that the login was unsuccessful. The TOE does not provide a reason for failure in the cases of a login failure.

6.3.4 FIA_UAU.7

81 For all authentication at the local CLI the TOE displays only "*" characters when the administrative password is entered so that the password is obscured.

6.3.5 FIA_AFL.1

82 The TOE is capable of tracking authentication failures of remote administrators.

83 When a user account has sequentially failed authentication the configured number of times, the account will be locked for a Security Administrator defined time period.

84 The administrator can configure the maximum number of failed attempts using the web GUI or CLI.

85 The TOE implements a password rescue account which permits administrators to (only) unlock locked administrators. This account is not subject to lockout.

6.3.6 FIA_X509_EXT.1/Rev

86 The TOE performs X.509 certificate validation at the following points:

a) On load of certificate responses

b) When processing OCSP responses

c) During IPsec peer authentication

d) When presented with a TLS client certificate for the purposes of user authentication

87 In all scenarios, certificates are checked for several validation characteristics:

a) If the certificate ‘notAfter’ date is in the past, then this is an expired certificate which is considered invalid;

b) The certificate chain must terminate with a trusted CA certificate;

c) A trusted CA certificate is defined as any certificate loaded into the TOE trust store that has, at a minimum, a basicConstraints extension with the CA flag set to TRUE;

d) The TOE validates the extendedKeyUsage field as follows:

i) TLS server certificates must have the Server authentication purpose in the extendedKeyUsage field

ii) TLS Client certificates must have the Client Authentication purpose in the extendedKeyUsage field

iii) OCSP certificates must have the OCSP signing purpose in the extendedKeyUsagefield

88 Certificate revocation checking is performed using OCSP.

89 The X.509 certificates for each of the given scenarios are validated using the certificate path validation algorithm defined in RFC 5280, which can be summarized as follows:

a) The public key algorithm and parameters are checked

b) The current date/time is checked against the validity period revocation status is checked

c) Issuer name of X matches the subject name of X+1

d) Name constraints are checked

e) Policy OIDs are checked

f) Policy constraints are checked; issuers are ensured to have CA signing bits

g) Path length is checked

h) Critical extensions are processed

90 If, during the entire trust chain verification activity, any certificate under review fails a verification check, then the entire trust chain is deemed untrusted.

6.3.7 FIA_X509_EXT.2

91 The TOE has a trust store where root CA and intermediate CA certificates can be stored. The trust store is not cached: if a certificate is deleted, it is immediately untrusted. If a certificate is added to the trust store, it is immediately trusted for its given scope.

92 Instructions for configuring the trusted IT entities to supply appropriate X.509 certificates are captured in the guidance documents.

93 As part of the verification process, OCSP is used to determine whether the certificate is revoked or not. If the OCSP server cannot be contacted, then the administrator has the option of choosing whether or not to accept the certificate.

6.3.8 FIA_X509_EXT.3

94 The TOE generates Certificate Request Messages and includes the following information: public key, common name, organization, organizational unit, country. Upon receiving the CA Certificate response, the TOE will validate the chain of certificates from the Root CA.

6.4 Security Management

6.4.1 FMT_MOF.1/ManualUpdate

95 The TOE restricts the ability to perform software updates to Security Administrators.

6.4.2 FMT_MOF.1/Functions

96 The TOE restricts the ability to modify (enable/disable) transmission of audit records to an external audit server to Security Administrators.

6.4.3 FMT_MTD.1/CoreData

97 Users are required to login before being provided with access to any administrative functions. Access to TSF data and functions, including managing the TOE's trust store, is restricted to Security Administrators as described by FMT_SMR.2 below.

6.4.4 FMT_SMR.2

98 The TOE implements role-based access control based on pre-defined profiles that are assigned when creating a user. The 'administrator' role is synonymous with the Security Administrator role defined in this document.

99 Management of TSF data via the CLI or web GUI is restricted to Security Administrators.

6.4.5 FMT_MTD.1/CryptoKeys

100 The TOE restricts the ability to manage SSH, TLS, IPsec and any configured X.509 private keys to Security Administrators.

6.4.6 FMT_SMF.1

101 The TOE may be managed via the CLI (console & SSH) or GUI (HTTPS). The specific management capabilities include:

a) Ability to administer the TOE locally and remotely

b) Ability to configure the access banner

c) Ability to configure the session inactivity time before session termination or locking

d) Ability to update the TOE and to verify the updates

e) Ability to configure the authentication failure parameters

f) Ability to configure audit behavior (enable/disable remote logging)

g) Ability to set the time which is used for time-stamps

h) Ability to manage the cryptographic keys, including import and management of X.509v3 certificates

6.5 Protection of the TSF

6.5.1 FPT_SKP_EXT.1

102 Keys are protected as described in Table 15. In all cases, plaintext keys cannot be viewed through an interface designed specifically for that purpose. Zeroization consists of a single pass overwrite of zeroes (0).

Table 15: Keys

Reference	Type	Generation and Use	Storage	Zeroization
Key Encryption Key (KEK)	Triple-DES (192 bits)	Hardcoded during manufacturing. Used only to protect keys stored in the flash, not for key transport.	Stored in Flash memory (plaintext).	Zeroized by using command 'write erase all'.
DRBG entropy input	SP800-90a CTR_DRBG (512 bits)	Entropy inputs to the DRBG function used to construct the DRBG seed.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
DRBG seed	SP800-90a CTR_DRBG (384-bits)	Input to the DRBG that determines the internal state of the DRBG. Generated using DRBG derivation function.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
DRBG Key	SP800-90a CTR_DRBG (256 bits)	This is the DRBG key used for SP800-90a CTR_DRBG.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
DRBG V	SP800-90a CTR_DRBG V (128 bits)	Internal V value used as part of SP800-90a CTR_DRBG	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module

Reference	Type	Generation and Use	Storage	Zeroization
Diffie-Hellman private key	Diffie-Hellman Group 14 (224 bits)	Generated internally during Diffie-Hellman Exchange. Used for establishing DH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
Diffie-Hellman public key	Diffie-Hellman Group 14 (2048 bits)	Generated internally during Diffie-Hellman Exchange. Used for establishing DH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
Diffie-Hellman shared secret	Diffie-Hellman Group 14 (2048 bits)	Established during Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
EC Diffie-Hellman private key	EC Diffie-Hellman (Curves: P-256 or P-384).	Generated internally during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
EC Diffie-Hellman public key	EC Diffie-Hellman (Curves: P-256 or P-384).	Generated internally during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
EC Diffie-Hellman shared secret	EC Diffie-Hellman (Curves: P-256 or P-384)	Established during EC Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
RADIUS server shared secret	8-128 characters shared secret	Entered by Security Administrator. Used for RADIUS server authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command ‘write erase all’ or by overwriting with a new secret
SKEYSEED	Shared Secret	A shared secret known only to IKE peers. It was	Stored in SDRAM	Zeroized by rebooting the module

Reference	Type	Generation and Use	Storage	Zeroization
	(160/256/384 bits)	derived via key derivation function defined in SP800-135 KDF (IKEv2) and it will be used for deriving IKE session authentication key.	memory (plaintext).	
IKE session authentication key	HMAC-SHA-1/256/384 (160/256/384 bits)	The IKE session (IKE Phase I) authentication key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv2). Used for IKEv2 payload integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
IKE session encryption key	Triple-DES (192 bits, 3 Key, CBC) /AES (128/192/256 bits, CBC)	The IKE session (IKE Phase I) encrypt key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv2). Used for IKE payload protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
IPSec session encryption key	Triple-DES (192 bits, 3 Key, CBC) / AES and AES-GCM (128/256 bits, CBC) NOTE: 192 bit CAVS tested, but not used.	The IPsec (IKE phase II) encryption key. This key is derived via a key derivation function defined in SP800-135 KDF (IKEv2). Used to secure IPsec traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
IPSec session authentication key	HMAC-SHA-1 (160 bits)	The IPsec (IKE Phase II) authentication key. This key is derived via using the KDF defined in SP800-135 KDF (IKEv2). Used to verify the	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module

Reference	Type	Generation and Use	Storage	Zeroization
		integrity of IPsec traffic.		
SSHv2 session key	AES (128/192/256 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used to secure SSHv2 traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
SSHv2 session authentication key	HMAC-SHA-1 (160-bit)	This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used to verify the integrity of SSHv2 traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
TLS pre-master secret	48 bytes secret	This key is transferred into the module, protected by TLS RSA public key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
TLS session encryption key	AES 128/192/256 bits	This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used to secure TLS traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
TLS session authentication key	HMAC-SHA-1/256/384 (160/256/384 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used to verify the integrity of TLS traffic.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
RSA Private Key	RSA 2048 bit private key	This key is generated in the module. Used for IKEv2, TLS, OCSP (signing OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command ‘write erase all’

Reference	Type	Generation and Use	Storage	Zeroization
RSA public key	RSA 2048 bits public key	This key is generated in the module. This Key can also be entered by the CO via SSH (CLI) and/or TLS (for the GUI). Used for IKEv2, TLS, OCSP (verifying OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	RSA public key
ECDSA Private Key	ECDSA suite B P-256 and P-384 curves	This key is generated in the module. Used for IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all'
ECDSA Public Key	ECDSA suite B P-256 and P-384 curves	This key is generated in the module. This Key can also be entered by the CO via SSH (CLI) and/or TLS (for the GUI). Used for IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (ciphertext) encrypted with KEK.	Zeroized by using command 'write erase all'.
Factory CA Public Key	RSA (2048 bits)	This is RSA public key. Loaded into the module during manufacturing. Used for Firmware verification.	Stored in Flash encrypted with KEK	Zeroized by using command 'write erase all'

6.5.2 FPT_APW_EXT.1

103 Passwords are protected as describe in Table 16. In all cases plaintext passwords cannot be viewed through an interface designed specifically for that purpose.

Table 16: Passwords

Reference	Type	Generation	Storage	Zeroization
Enable secret	8-32 characters password	Entered by Security Administrator.	Stored in Flash memory (ciphertext) encrypted with KEK	Zeroized by using command 'write erase all' or by overwriting with a new secret
User Password	8-32 characters password	Entered by Security Administrator.	Stored in Flash memory (ciphertext) encrypted with KEK	Zeroized by using command 'write erase all' or by overwriting with a new secret

6.5.3 FPT_TST_EXT.1

104 The TOE runs a suite of self-tests during power-up and periodically during operation, which includes demonstration of the correct operation of the hardware and the use of cryptographic functions to verify the integrity of TSF executable code and static data. An administrator can choose to reboot the TOE to perform power-up self-tests. The Mobility Controller runs the suite of FIPS 140-2 validated cryptographic module self-tests during start-up or on request from the administrator, including immediately after generation of a key (FIPS self-tests, including the continuous RNG test).

105 The following test are performed:

a) ArubaOS OpenSSL Module:
   i) Algorithm Known Answer Tests
   ii) ECDSA (sign/verify)

b) ArubaOS Cryptographic Module
   i) Algorithm Known Answer Tests
   ii) RSA (sign/verify)
   iii) ECDSA (sign/verify)

c) ArubaOS Uboot BootLoader Module
   i) Firmware Integrity Test: RSA 2048-bit Signature Validation

d) Aruba Hardware Known Answer Tests

106 The following Conditional Self-tests are performed by the TOE:

a) **Continuous Random Number Generator Test.** This test is run upon generation of random data by the switch's random number generators to detect failure to a constant value. The module stores the first random number for subsequent comparison, and the module compares the value of the new random number with the random number generated in the previous round and enters an error state if the comparison is successful.

b) **Bypass test.** Ensures that the system has not been placed into a mode of operation where cryptographic operations have been bypassed, without the explicit configuration of the cryptographic officer. To conduct the test, a SHA1

hash of the configuration file is calculated and compared to the last known good hash of the configuration file. If the hashes match, the test is passed. Otherwise, the test fails (indicating possible tampering with the configuration file) and the system is halted.

c) **RSA Pairwise Consistency test.** When the TOE generates a public and private key pair, it carries out pair-wise consistency tests for both encryption and digital signing. The test involves encrypting a randomly-generated message with the public key. If the output is equal to the input message, the test fails. The encrypted message is then decrypted using the private key and if the output is not equal to the original message, the test fails. The same random message is then signed using the private key and then verified with the public key. If the verification fails, the test fails.

d) **ECDSA Pairwise Consistency test.** See above RSA pairwise consistency test description.

e) **Firmware Load Test.** This test is identical to the Uboot BootLoader Module Firmware Integrity Test, except that it is performed at the time a new software image is loaded onto the system. Instead of being performed by the BootLoader, the test is performed by the ArubaOS operating system. If the test fails, the newly loaded software image will not be copied into the image partition, and instead will be deleted.

107 Known-answer tests (KAT) involve operating the cryptographic algorithm on data for which the correct output is already known and comparing the calculated output with the previously generated output (the known answer). If the calculated output does not equal the known answer, the known-answer test shall fail.

108 If a self-test fails, the TOE will immediately halt operation and enter an error state thereby preventing potentially insecure operations (i.e., maintaining a secure state). The controller will reboot after a self-test failure. During reboot, memory is re-initialized, which wipes all keys and user data. If a self-test failure continues to occur, the controller will continue to reboot repeatedly and will require return to manufacturer.

109 The above tests are sufficient to demonstrate that the TSF is operating correctly by verifying the integrity of the TSF and the correct operation of cryptographic components.

6.5.4 FPT_TUD_EXT.1

110 The TOE allows administrators to query the current version of its firmware/software and allows those administrators to manually initiate firmware/software updates. Prior to installing any update, the administrator can verify the digital signature of the update.

111 Administrators can update the TOE executable code using image files manually downloaded from the Aruba support portal. The administrator may perform an update from either the WebUI or CLI. Upgrade instructions are documented in the release notes for each software release, which will be posted in the same directory as the image file on the support portal.

112 A SHA-256 hash of each update image is digitally signed using Aruba's code signing certificate (RSA 2048 bit). The signing certificate is issued by Aruba's internal CA. Aruba's code signing public key is programmed into the Boot ROM of all Aruba products at the time of manufacturing.

113 When an update is initiated, the TOE verifies the digital signature with the stored public key (stored in Boot ROM). Upon successful verification, the TOE boots using

the new image. Should verification fail, the TOE will enter into an error state. The TOE's error state will allow direct console access only, where an administrator can change to a new file partition.

6.5.5 FPT_STM_EXT.1

114 The TOE has an internal battery-backed hardware clock that provides reliable time stamps used for auditing. The internal clock may be synchronized with a time signal obtained from an external NTP server.

115 The TOE makes used of time for the following:

a) Audit record timestamps

b) Session timeouts (lockout enforcement)

c) Certificate validation

6.6 TOE Access

6.6.1 FTA_SSL_EXT.1

116 The Security Administrator may configure the TOE to terminate an inactive local interactive session (CLI) following a specified period of time.

6.6.2 FTA_SSL.3

117 The Security Administrator may configure the TOE to terminate an inactive remote interactive session (CLI and Web UI) following a specified period of time.

6.6.3 FTA_SSL.4

118 Administrative users may terminate their own sessions at any time.

6.6.4 FTA_TAB.1

119 The TOE displays an administrator configurable message to users prior to login at the CLI and web GUI.

6.7 Trusted Path/Channels

6.7.1 FTP_ITC.1

120 The TOE supports secure communication with the following IT entities:

a) Audit server via IPsec

b) Authentication server via IPsec

c) Mobility Controllers via IPsec

6.7.2 FTP_TRP.1/Admin

121 The TOE provides the following trusted paths for remote administration:

a) CLI over SSH

b) Web GUI over HTTPS

7 Rationale

7.1 Conformance Claim Rationale

122 The following rationale is presented with regard to the PP conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is network device, consistent with the NDcPP.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the NDcPP.

c) **Security objectives.** As shown in section 4, the security objectives are reproduced directly from the NDcPP.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced directly from the NDcPP. No additional requirements have been specified.

7.2 Security Objectives Rationale

123 All security objectives are drawn directly from the NDcPP.

7.3 Security Requirements Rationale

124 All security requirements are drawn directly from the NDcPP. Table 17 presents a mapping between threats and SFRs as presented in the NDcPP.

Table 17: NDcPP SFR Rationale

Identifier	SFR Rationale
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	• The Administrator role is defined in FMT_SMR.2 and the relevant administration capabilities are defined in FMT_SMF.1 and FMT_MTD.1/CoreData, with optional additional capabilities in FMT_MOF.1/Services and FMT_MOF.1/Functions • The actions allowed before authentication of an Administrator are constrained by FIA_UIA_EXT.1, and include the advisory notice and consent warning message displayed according to FTA_TAB.1 • The requirement for the Administrator authentication process is described in FIA_UAU_EXT.2 • Locking of Administrator sessions is ensured by FTA_SSL_EXT.1 (for local sessions), FTA_SSL.3 (for remote sessions), and FTA_SSL.4 (for all interactive sessions) • The secure channel used for remote Administrator connections is specified in FTP_TRP.1/Admin • (Malicious actions carried out from an Administrator session are separately addressed by T.UNDETECTED_ACTIVITY)

Identifier	SFR Rationale
	• (Protection of the Administrator credentials is separately addressed by T.PASSWORD_CRACKING).
T.WEAK_CRYPTOGRAPHY	• Requirements for key generation and key distribution are set in FCS_CKM.1 and FCS_CKM.2 respectively • Requirements for use of cryptographic schemes are set in FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash • Requirements for random bit generation to support key generation and secure protocols (see SFRs resulting from T.UNTRUSTED_COMMUNICATION_CHANNELS) are set in FCS_RBG_EXT.1 • Management of cryptographic functions is specified in FMT_SMF.1
T.UNTRUSTED_COMMUNICATION_CHANNELS	• The general use of secure protocols for identified communication channels is described at the top level in FTP_ITC.1 and FTP_TRP.1/Admin; for distributed TOEs the requirements for inter-component communications are addressed by the requirements in FPT_ITT.1 • Requirements for the use of secure communication protocols are set for all the allowed protocols in FCS_DTLSC_EXT.1, FCS_DTLSC_EXT.2, FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2, FCS_HTTPS_EXT.1, FCS_IPSEC_EXT.1, FCS_SSHC_EXT.1, FCS_SSHS_EXT.1, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2 • Optional and selection-based requirements for use of public key certificates to support secure protocols are defined in FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3
T.WEAK_AUTHENTICATION_ENDPOINTS	• The use of appropriate secure protocols to provide authentication of endpoints (as in the SFRs addressing T.UNTRUSTED_COMMUNICATION_CHANNELS) are ensured by the requirements in FTP_ITC.1 and FTP_TRP.1/Admin; for distributed TOEs the authentication requirements for endpoints in inter-component communications are addressed by the requirements in FPT_ITT.1 • Additional possible special cases of secure authentication during registration of distributed TOE components are addressed by FCO_CPC_EXT.1 and FTP_TRP.1/Join.
T.UPDATE_COMPROMISE	• Requirements for protection of updates are set in FPT_TUD_EXT.1 • Additional optional use of certificate-based protection of signatures can be specified using FPT_TUD_EXT.2, supported by the X.509 certificate processing requirements in FIA_X509_EXT.1, FIA_X509_EXT.2 and FIA_X509_EXT.3

Identifier	SFR Rationale
	• Requirements for management of updates are defined in FMT_SMF.1 and (for manual updates) in FMT_MOF.1/ManualUpdate, with optional requirements for automatic updates in FMT_MOF.1/AutoUpdate
T.UNDETECTED_ACTIVITY	• Requirements for basic auditing capabilities are specified in FAU_GEN.1 and FAU_GEN.2, with timestamps provided according to FPT_STM_EXT.1 • Requirements for protecting audit records stored on the TOE are specified in FAU_STG.1 • Requirements for secure transmission of local audit records to an external IT entity via a secure channel are specified in FAU_STG_EXT.1 • Optional additional requirements for dealing with potential loss of locally stored audit records are specified in FAU_STG_EXT.2/LocSpace, and FAU_STG.3/LocSpace • If (optionally) configuration of the audit functionality is provided by the TOE then this is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MOF.1/Functions.
T.SECURITY_FUNCTIONALITY_COMPROMISE	• Protection of secret/private keys against compromise is specified in FPT_SKP_EXT.1 • Secure destruction of keys is specified in FCS_CKM.4 • If (optionally) management of keys is provided by the TOE then this is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MTD.1/CryptoKeys • (Protection of passwords is separately covered under T.PASSWORD_CRACKING)
T.PASSWORD_CRACKING	• Requirements for password lengths and available characters are set in FIA_PMG_EXT.1 • Protection of password entry by providing only obscured feedback is specified in FIA_UAU.7 • Actions on reaching a threshold number of consecutive password failures are specified in FIA_AFL.1 • Requirements for secure storage of passwords are set in FPT_APW_EXT.1.
T.SECURITY_FUNCTIONALITY_FAILURE	• Requirements for running self-test(s) are defined in FPT_TST_EXT.1 • Optional use of certificates to support self-test(s) is defined in FPT_TST_EXT.2 (with support for the use of certificates in FIA_X509_EXT.1, FIA_X509_EXT.2, and FIA_X509_EXT.3),

Annex A: Extended Components Definition

125 This annex reproduces the NDcPP Appendix C extended components definition.

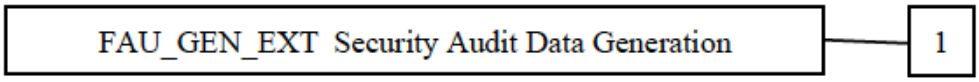
Security Audit (FAU)

Security Audit Data Generation (FAU_GEN_EXT)

Family Behaviour

This component defines the requirements for components in a distributed TOE to generate security audit data.

Component levelling



FAU_GEN_EXT.1 Security audit data shall be generated by all components in a distributed TOE

Management: FAU_GEN_EXT.1

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

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_GEN_EXT.1

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

a) No audit necessary.

FAU_ GEN_EXT.1 Security Audit Data Generation for Distributed TOE Components

FAU_GEN_EXT.1 Security Audit Data Generation

Hierarchical to: No other components.

Dependencies: None.

FAU_GEN_EXT.1.1. The TSF shall be able to generate audit records for each TOE component. The audit records generated by the TSF of each TOE component shall include the subset of security relevant audit events which can occur on the TOE component.

Application Note 135

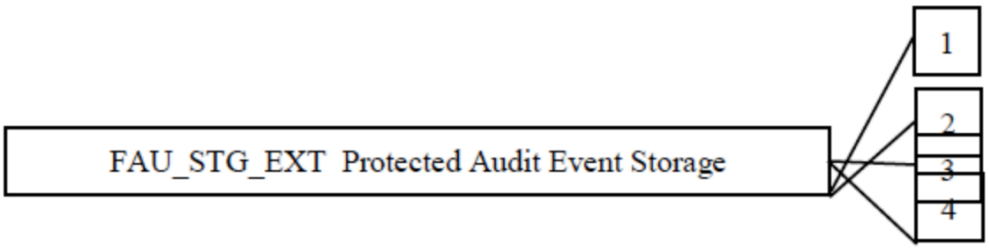
The TOE must be able to generate audit records for each TOE component. Some TOE components of a distributed TOE might not implement the complete TSF of the overall TOE but only a subset of the TSF. The audit records for each TOE component need to cover all security relevant audit events according to the subset of the TSF implemented by this particular TOE component but not necessarily all security relevant audit events according to the TSF of the overall TOE. If a security-relevant event can occur on multiple TOE components, it needs to cause generation of an audit record uniquely identifying the component associated with the event. The ST author shall identify for each TOE component which of the overall required audit events defined in FAU_GEN.1.1 are logged. The ST author may decide to do this by providing a corresponding table. The information provided needs to be in agreement with Table 1. The overall TOE needs to cover all auditable events listed in Table 2 (and Tables 4 and 5 as applicable to the overall TOE).

Protected audit event storage (FAU_STG_EXT)

Family Behaviour

This component defines the requirements for the TSF to be able to securely transmit audit data between the TOE and an external IT entity.

Component levelling



FAU_STG_EXT.1 Protected audit event storage requires the TSF to use a trusted channel implementing a secure protocol.
FAU_STG_EXT.2 Counting lost audit data requires the TSF to provide information about audit records affected when the audit log becomes full.
FAU_STG_EXT.3 Protected Local audit event storage for distributed TOEs requires the TSF to use a trusted channel to protect audit transfer to another TOE component.
FAU_STG_EXT.4 Protected Remote audit event storage for distributed TOEs requires the TSF to use a trusted channel to protect audit transfer to another TOE component.

Management: FAU_STG_EXT.1, FAU_STG_EXT.2, FAU_STG_EXT.3, FAU_STG_EXT.4

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

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_STG_EXT.1, FAU_STG_EXT.2, FAU_STG_EXT.3, FAU_STG_EXT.4

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

a) No audit necessary.

FAU_ STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1 Protected Audit Event 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.

Application Note 136
For selecting the option of transmission of generated audit data to an external IT entity the TOE 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. Since the external audit server is not part of the TOE, there are no requirements on it except the capabilities for ITC transport for audit data. No requirements are placed upon the format or underlying protocol of the audit data being transferred. The TOE must be capable of being configured to transfer audit data to an external IT entity without Administrator intervention. Manual transfer would not meet the requirements. Transmission could be done in real-time or periodically. If the transmission is not done in real-time then the TSS describes what event stimulates the transmission to be made and what range of frequencies the TOE supports for making transfers of audit data to the audit server; the TSS also suggests typical acceptable frequencies for the transfer.
For distributed TOEs each component must be able to export audit data across a protected channel external (FTP_ITC.1) or intercomponent (FPT_ITT.1 or FTP_ITC.1) as appropriate. At least one component of the TOE must be able to export audit records via FTP_ITC.1 such that all TOE audit records can be exported to an external IT entity.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself.
[selection:

- TOE shall consist of a single standalone component that stores audit data locally,
- The TOE shall be a distributed TOE that stores audit data on the following TOE components: [assignment: identification of TOE components],
- The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [assignment: list of TOE components that do not store audit data locally and the other TOE components to which they transmit their generated audit data].

Application Note 137
If the TOE is a standalone TOE (i.e. not a distributed TOE) the option 'The TOE shall consist of a single standalone component that stores audit data locally' shall be selected.
If the TOE is a distributed TOE the option 'The TOE shall be a distributed TOE that stores audit data on the following TOE components: [assignment: identification of TOE components]' shall be selected and the TOE components which store audit data locally shall be listed in the assignment.

Since all TOEs are required to provide functions to store audit data locally this option needs to be selected for all distributed TOEs. In addition, FAU_GEN_EXT.1 and FAU_STG_EXT.3 shall be claimed in the ST. If the distributed TOE consists only of components which are storing audit data locally, it is sufficient to select only the option 'The TOE shall be a distributed TOE that stores audit data on the following TOE components: [assignment: identification of TOE components]' and add FAU_GEN_EXT.1 and FAU_STG_EXT.3.
If the TOE is a distributed TOE and some TOE components are not storing audit data locally, the option 'The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [assignment: list of TOE components that do not store audit data locally and the other TOE components to which they transmit their generated audit data]' shall be selected in addition to the option 'The TOE shall be a distributed TOE that stores audit data on the following TOE components: [assignment: identification of TOE components]'. In that case FAU_STG_EXT.4 shall be claimed in the ST in addition to FAU_GEN_EXT.1 and FAU_STG_EXT.3. For the option 'The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [assignment: list of TOE components that do not store audit data locally and the other TOE components to which they transmit their generated audit data]' the TOE components that to not store audit data locally shall be mapped to the TOE components to which they transmit their generated audit data.
For distributed TOEs this SFR can be fulfilled either by every TOE component storing its own security audit data locally or by one or more TOE components storing audit data locally and other TOE components which are not storing audit information locally sending security audit data to other TOE components for local storage. For the transfer of security audit data between TOE components a protected channel according to FTP_ITC.1 or FPT_ITT.1 shall be used. The TSS shall describe which TOE components store security audit data locally and which TOE components do not store security audit data locally. For the latter, the TSS shall describe at which other TOE component the audit data is stored locally.
FAU_STG_EXT.1.3 The TSF shall [selection: drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]] when the local storage space for audit data is full.
Application Note 138
The external log server might be used as alternative storage space in case the local storage space is full. The “other action” could in this case be defined as “send the new audit data to an external IT entity”.
For distributed TOEs each component is not required to store generated audit data locally but the overall TOE needs to be able to store audit data locally. Each component must at least provide the ability to temporarily buffer audit information locally to ensure that audit records are preserved in case of network connectivity issues. Buffering audit information locally, does not necessarily involve non-volatile memory: audit information could be buffered in volatile memory. However, the local storage of audit information in the sense of FAU_STG_EXT.1.3 needs to be done in non-volatile memory. For every component which performs local storage of audit information, the behaviour when local storage is exhausted needs to be described. For every component which is buffering audit information instead of storing audit information locally itself, it needs to be described what happens in case the buffer space is exhausted.

FAU_ STG_EXT.2 Counting lost audit data

FAU_STG_EXT.2 Counting lost audit data

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.2.1 The TSF shall provide information about the number of [selection: dropped, overwritten, [assignment: other information]] audit records in the case where the local storage has been filled and the TSF takes one of the actions defined in FAU_STG_EXT.1.3.

Application Note 139
This option should be chosen if the TOE supports this functionality.
In case the local storage for audit records is cleared by the Administrator, the counters associated with the selection in the SFR should be reset to their initial value (most likely to 0). The guidance documentation should contain a warning for the Administrator about the loss of audit data when he clears the local storage for audit records.
For distributed TOEs each component that implements counting of lost audit data has to provide a mechanism for Administrator access to, and management of, this information.
If FAU_STG_EXT.2 is added to the ST, the ST has to make clear any situations in which lost audit data is not counted.

FAU_ STG_EXT.3 Protected Local Audit Event Storage for Distributed TOEs

FAU_STG_EXT.3 Protected Audit Event Storage

Hierarchical to: No other components.

Dependencies: FAU_GEN_EXT.1 Security Audit data generation for Distributed TOE Components [FPT_ITT.1 Intra-TSF Trusted Channel or FTP_ITC.1 Inter-TSF Trusted Channel]

FAU_STG_EXT.3.1 The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: [assignment: table of components and for each component its action chosen according to the following: [selection: drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]]].

Application Note 140
If a component of a distributed TOE collects data from other components and then forwards it to another component or external IT entity (cf. FAU_STG_EXT.1.1) then the operations in this SFR must be performed in a way to cover the storage space action(s) for all of the audit data that the TOE collects (i.e. not just for the data generated by the collecting component for itself).
It is acceptable for a TOE component to store audit information in multiple places (e.g. for redundancy), whether locally in the TOE component itself and in another TOE component, or in more than one other TOE component.
TOE components are not required to monitor or audit connectivity or network outages between TOE components. This aspect is covered by the assumption A.COMPONENTS_RUNNING.

FAU_ STG_EXT.4 Protected Remote Audit Event Storage for Distributed TOEs

FAU_STG_EXT.4 Protected Audit Event Storage

Hierarchical to: No other components.

Dependencies:	FAU_GEN_EXT.1 Security Audit data generation for Distributed TOE Components [FPT_ITT.1 Intra-TSF Trusted Channel or FTP_ITC.1 Inter-TSF Trusted Channel]
FAU_STG_EXT.4.1	Each TOE component which does not store security audit data locally shall be able to buffer security audit data locally until it has been transferred to another TOE component that stores or forwards it. All transfer of audit records between TOE components shall use a protected channel according to [selection: FPT_ITT.1, FTP_ITC.1].

Application Note 141
If a component of a distributed TOE collects data from other components and then forwards it to another component or external IT entity (cf. FAU_STG_EXT.1.1) then the operations in this SFR must be performed in a way to cover the storage space action(s) for all of the audit data that the TOE collects (i.e. not just for the data generated by the collecting component for itself).
It is acceptable for a TOE component to store audit information in multiple places (e.g. for redundancy), whether locally in the TOE component itself and in another TOE component, or in more than one other TOE component.
TOE components are not required to monitor or audit connectivity or network outages between TOE components. This aspect is covered by the assumption A.COMPONENTS_RUNNING.

Cryptographic Support (FCS)

Random Bit Generation (FCS_RBG_EXT)

FCS_RBG_EXT.1 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

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: FCS_RBG_EXT.1
The following actions could be considered for the management functions in FMT:
a) There are no management activities foreseen
Audit: FCS_RBG_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) Minimal: failure of the randomization process

FCS_RBG_EXT.1 Random Bit Generation

Hierarchical to:	No other components

Dependencies: No other components

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 (AES)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: [assignment: number of software-based sources] software-based noise source, [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 142
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 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-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based 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 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.

Cryptographic Protocols (FCS_DTLSC_EXT, FCS_DTLSS_EXT, FCS_HTTPS_EXT, FCS_IPSEC_EXT, FCS_NTP_EXT, FCS_SSHC_EXT, FCS_SSHS_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT)

FCS_DTLSC_EXT 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:
a) 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:
a) Failure of DTLS session establishment
b) DTLS session establishment
c) DTLS session termination

FCS_DTLSC_EXT.1 DTLS Client Protocol

Hierarchical to:	No other components
Dependencies:	FCS_CKM. 1DataEncryption1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen1SigGen 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.1.1	The TSF shall implement [selection: DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)] supporting the following ciphersuites: [assignment: List of optional ciphersuites and reference to RFC in which each is defined]

Application Note 143
The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported.
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.1.2	The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125 section 6.

Application Note 144
The rules for verification of identity are described in Section 6 of RFC 6125. 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 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 DTLS 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 is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or

Subject Alternative name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the evaluation activity.

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 145
"Revocation status" refers to a 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.
If DTLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT.

FCS_DTLSC_EXT.1.4 The TSF shall [selection: not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves] in the Client Hello.

Application Note 146
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_DTLS_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.

FCS_DTLSC_EXT.2 DTLS Client Protocol with Authentication

Hierarchical to: FCS_DTLSC_EXT.1 DTLS Client Protocol

Dependencies: FCS_CKM.1/DataEncryption Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (AES 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.2.1 The TSF shall implement [selection: DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)] supporting the following ciphersuites:

- [assignment: List of optional ciphersuites and reference to RFC in which each is defined].

Application Note 147
The ST author should select the ciphersuites that are supported.
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.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125 section 6.

Application Note 148
The rules for verification of identity are described in Section 6 of RFC 6125. 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 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 DTLS server's certificate.

FCS_DTLSC_EXT.2.3 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 149
"Revocation status" refers to a 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.
If DTLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT.

FCS_DTLSC_EXT.2.4 The TSF shall [selection: not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves] in the Client Hello].

Application Note 150
If ciphersuites with elliptic curves were selected in FCS_DTLSC_EXT.2.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_DTLS_EXT.2.1, 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.

FCS_DTLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

Application Note 151
The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a DTLS server for DTLS mutual authentication.

FCS_DTLSC_EXT.2.6 The TSF shall [selection: terminate the DTLS session, silently discard the record] if a message received contains an invalid MAC.

Application Note 152
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_DTLSC_EXT.2.7 The TSF shall detect and silently discard replayed messages for:

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

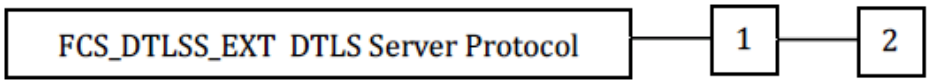
Application Note 153
Replay Detection is described in section 4.1.2.6 of DTLS 1.2 (RFC 6347) and section 4.1.2.5 of DTLS 1.0 (RFC 4347). For each received record, the receiver verifies the record contains a sequence number is within the sliding receive window and does not duplicate the sequence number of any other record received during the session.
"Silently Discard" means the TOE discards the packet responding.

FCS_DTLSS_EXT 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 the 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:
a) 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:
a) Failure of DTLS session establishment.
b) DTLS session establishment

c) DTLS session termination

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 (AES 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.1.1 The TSF shall implement [selection: DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)] supporting the following ciphersuites:

- [assignment: List of optional ciphersuites and reference to RFC in which each is defined]

Application Note 154
The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported.
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 [assignment: list of protocol versions].

Application Note 155
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 156
The process to validate the IP address of a 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 [selection: perform RSA key establishment with key size [selection: 2048 bits, 3072 bits, 4096 bits]; generate EC Diffie-Hellman parameters over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie-Hellman parameters of size [selection: 2048 bits, 3072 bits]].

Application Note 157

If the ST lists a DHE or ECDHE ciphersuite in FCS_DTLSS_EXT.1.1, the ST must include the Diffie-Hellman or NIST curves selection in the requirement. FMT_SMF.1 requires the configuration of the key agreement parameters in order 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 158
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 159
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 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.2 DTLS Server Protocol with mutual authentication

Hierarchical to: FCS_DTLSS_EXT.1 DTLS Server Protocol

Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1//DataEncryption Cryptographic operation (AES 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.2.1 The TSF shall implement [selection: DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)] supporting the following ciphersuites:

- [assignment: List of optional ciphersuites and reference to RFC in which each is defined].

Application Note 160
The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported.
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.2.2 The TSF shall deny connections from clients requesting [assignment: list of protocol versions].

Application Note 161

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

Application Note 162
The process to validate the IP address of a 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.2.4 The TSF shall [selection: perform RSA key establishment with key size [selection: 2048 bits, 3072 bits, 4096 bits]; generate EC Diffie-Hellman parameters over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie-Hellman parameters of size [selection: 2048 bits, 3072 bits]].

Application Note 163
If the ST lists a DHE or ECDHE ciphersuite in FCS_DTLSS_EXT.2.1, the ST must include the Diffie-Hellman or NIST curves selection in the requirement. FMT_SMF.1 requires the configuration of the key agreement parameters in order to establish the security strength of the DTLS connection.
FCS_DTLSS_EXT.2.5 The TSF shall [selection: terminate the DTLS session, silently discard the record] if a message received contains an invalid MAC.
Application Note 164
The Message Authentication Code (MAC) is negotiated during the 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.2.6 The TSF shall detect and silently discard replayed messages for:

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

Application Note 165
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 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.2.7 The TSF shall support mutual authentication of DTLS clients using X.509v3 certificates.

Application Note 166
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.2.8 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 167
"Revocation status" refers to a 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.
If DTLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT.

FCS_DTLSS_EXT.2.9 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 168
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.

FCS_HTTPS_EXT.1 HTTPS Protocol

Family Behaviour

Components in this family define the 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

FCS_HTTPS_EXT.1 HTTPS requires that HTTPS be implemented according to RFC 2818 and supports TLS.
Management: FCS_HTTPS_EXT.1
The following actions could be considered for the management functions in FMT:
a) There are no management activities foreseen.
Audit: FCS_HTTPS_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) There are no auditable events foreseen.

FCS_HTTPS_EXT.1 HTTPS Protocol

Hierarchical to: No other components

Dependencies: [FCS_TLSC_EXT.1 TLS Client Protocol, or FCS_TLSS_EXT.1 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 the HTTPS protocol using TLS.

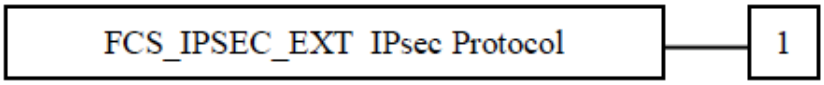
FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [selection: not establish the connection, request authorization to establish the connection, [assignment: other action]] if the peer certificate is deemed invalid.

FCS_IPSEC_EXT.1 IPsec Protocol

Family Behaviour

Components in this family address the requirements for protecting communications using IPsec.

Component levelling



FCS_IPSEC_EXT.1 IPsec requires that IPsec be implemented as specified.
Management: FCS_IPSEC_EXT.1
The following actions could be considered for the management functions in FMT:
a) Maintenance of SA lifetime configuration
Audit: FCS_IPSEC_EXT.1
The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:
a) Decisions to DISCARD, BYPASS, PROTECT network packets processed by the TOE.
b) Failure to establish an IPsec SA
c) IPsec SA establishment
d) IPsec SA termination
e) Negotiation “down” from an IKEv2 to IKEv1 exchange.

FCS_IPSEC_EXT.1 Internet Protocol Security (IPsec) Communications

Hierarchical to: No other components

Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (AES 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.

Application Note 169
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: tunnel mode, transport 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, AES-CBC-192, AES-CBC-256 (specified in RFC 3602), no other algorithm] together with a Secure Hash Algorithm (SHA)-based HMAC [selection: HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, no other algorithm] and [selection: AES-GCM-128, AES-GCM-192, AES-GCM-256 (specified in RFC 4106), no other 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)].

Application Note 170
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 171
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 172
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 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.

Application Note 173
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 it 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 174
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: 14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP), 24 (2048-bit MODP with 256-bit POS)].

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 175
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].

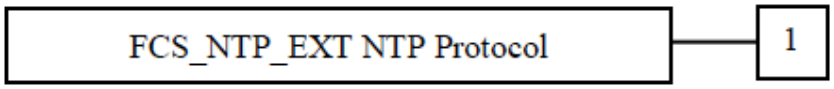
FCS_IPSEC_EXT.1.14 The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: [selection: SAN: IP address, SAN: Fully Qualified 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]].

FCS_NTP_EXT.1 NTP Protocol

Family Behaviour

The component in this family addresses the ability for a TOE to protect NTP time synchronization traffic.

Component levelling



FCS_NTP_EXT.1 Requires NTP to be implemented as specified
Management: FCS_NTP_EXT.1
The following actions could be considered for the management functions in FMT:
a) Ability to configure NTP
Audit: FCS_NTP_EXT.1
The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:
a) No audit requirements are specified.

FCS_NTP_EXT.1 NTP Protocol

Hierarchical to: No other components

Dependencies: FCS_COP.1 Cryptographic operation
[FCS_DTLSC_EXT.1 DTLC Client Protocol or
FCS_IPSEC_EXT.1 IPsec Protocol]

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s) [selection: NTP v3 (RFC 1305), NTP v4 (RFC 5905)].

FCS_NTP_EXT.1.2 The TSF shall update its system time using [selection:

- Authentication using [selection: SHA1, SHA256, SHA384, SHA512, AES-CBC-128, AES-CBC-256] as the message digest algorithm(s);
- [selection: IPsec, DTLS] to provide trusted communication between itself and an NTP time source.

].

FCS_NTP_EXT.1.3 The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

Application Note 176
The broadcast and multicast addresses are deemed as any addressing scheme designed to be one-to-many.

FCS_NTP_EXT.1.4 The TSF shall support configuration of at least three (3) NTP time sources.

Application Note 177
The TOE has to support configuration of at least three (3) time sources though not mandated that the TOE is configured to always use at least 3 time sources.

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:
a) 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:
a) Failure of SSH session establishment
b) SSH session establishment
c) SSH session termination

FCS_SSHC_EXT.1 SSH Client Protocol

Hierarchical to: No other components

Dependencies: FCS_CKM.1Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (AES 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 that complies with RFC(s) [selection: 4251, 4252, 4253, 4254, 5647, 5656, 6187, 6668, 8332].

Application Note 178
The ST author selects which of the RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). 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.
RFC 5647 only applies to the RFC compliant implementation of GCM; a TOE that only implements the "@openssh.com" variant of GCM should not select 5647. 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).

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 179
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: [assignment: list of encryption algorithms].

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 [assignment: list of data integrity MAC algorithms] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7 The TSF shall ensure that [assignment: list of key exchange 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 no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

Application Note 180
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 total

incoming and outgoing data needs to be counted. The rekey 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 or [selection: a list of trusted certification authorities, no other methods] as described in RFC 4251 section 4.1.

Application Note 181
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.

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:
a) There are no management activities foreseen.
Audit: FCS_SSHS_EXT.1
The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:
a) Failure of SSH session establishment
b) SSH session establishment
c) SSH session termination

FCS_SSHS_EXT.1 SSH Server Protocol

Hierarchical to: No other components

Dependencies: FCS_CKM.1Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (AES 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 that complies with RFC(s) [selection: 4251, 4252, 4253, 4254, 5647, 5656, 6187, 6668, 8332].

Application Note 182
The ST author selects which of the RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). 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.
RFC 5647 only applies to the RFC compliant implementation of GCM; a TOE that only implements the "@openssh.com" variant of GCM should not select 5647. 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).

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, password-based.

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 183
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: [assignment: encryption algorithms].

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 [assignment: list of MAC algorithms] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [assignment: list of key exchange 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 no more

than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

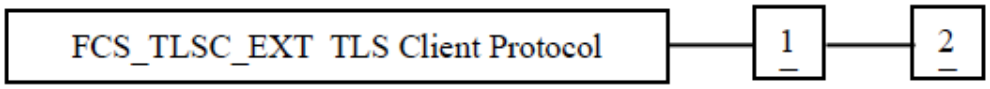
Application Note 184
This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold, the total incoming and outgoing data needs to be counted. The rekey 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_TLSC_EXT 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:
a) 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:
a) Failure of TLS session establishment
b) TLS session establishment
c) TLS session termination

FCS_TLSC_EXT.1 TLS 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 (AES 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.1.1 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:

- [assignment: list of optional ciphersuites and reference to RFC in which each is defined].

Application Note 185

The ciphersuites to be tested in the evaluated configuration are limited by this requirement.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

Application Note 186
The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (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 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 is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or Subject Alternative name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the evaluation activity.

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 187
"Revocation status" refers to a 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 TLS is selected in FTP_ITC then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev.

If TLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT.

FCS_TLSC_EXT.1.4 The TSF shall [selection: not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves] in the Client Hello.

Application Note 188
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_TLS_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.

FCS_TLSC_EXT.2 TLS Client Protocol with Authentication

Hierarchical to: FCS_TLSC_EXT.1 TLS Client Protocol

Dependencies: FCS_CKM.1Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (AES 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.2.1 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:

- [assignment: list of optional ciphersuites and reference to RFC in which each is defined].

Application Note 189
The ciphersuites to be tested in the evaluated configuration are limited by this requirement.

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

Application Note 190
The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (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 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 is required for the purposes of backwards compatibility. Additionally, support for use of IP addresses in the Subject Name or Subject Alternative name is discouraged as against best practices but may be implemented. Finally, the client should avoid constructing reference identifiers using wildcards. However, if the presented identifiers include wildcards, the client must follow the best practices regarding matching; these best practices are captured in the evaluation activity.

FCS_TLSC_EXT.2.3 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 191
"Revocation status" refers to a 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 TLS is selected in FTP_ITC then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev.
If TLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT.

FCS_TLSC_EXT.2.4 The TSF shall [selection: not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves] in the Client Hello.

Application Note 192
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_TLS_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.

FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

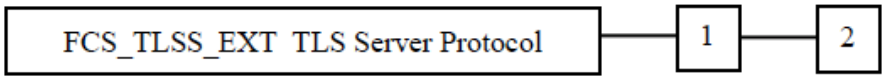
Application Note 193
The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

FCS_TLSS_EXT 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:
a) 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:
a) Failure of TLS session establishment
b) TLS session establishment
c) TLS session termination

FCS_TLSS_EXT.1 TLS 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 (AES 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.1.1 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:

- [assignment: list of optional ciphersuites and reference to RFC in which each is defined].

Application Note 194
The ciphersuites to be tested in the evaluated configuration are limited by this requirement.

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 195
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 [selection: perform RSA key establishment with key size [selection: 2048 bits, 3072 bits, 4096 bits]; generate EC Diffie-Hellman parameters over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie-Hellman parameters of size [selection: 2048 bits, 3072 bits]].

Application Note 196
The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.1.1.

FCS_TLSS_EXT.2 TLS Server Protocol with mutual authentication

Hierarchical to: FCS_TLSS_EXT.1 TLS Server Protocol

Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (AES 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.2.1 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:

- [assignment: list of optional ciphersuites and reference to RFC in which each is defined].

Application Note 197
The ciphersuites to be tested in the evaluated configuration are limited by this requirement.

FCS_TLSS_EXT.2.2 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 198
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.2.3 The TSF shall [selection: perform RSA key establishment with key size [selection: 2048 bits, 3072 bits, 4096 bits]; generate EC Diffie-Hellman parameters over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie-Hellman parameters of size [selection: 2048 bits, 3072 bits]].

Application Note 199
The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.2.1.

FCS_TLSS_EXT.2.4 The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

Application Note 200
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.

FCS_TLSS_EXT.2.5 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 201
"Revocation status" refers to a 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 TLS is selected in FTP_ITC then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev.
If TLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT.

FCS_TLSS_EXT.2.6 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 202
This requirement only applies to those TOEs performing mutually-authenticated TLS (FCS_TLSS_EXT.2.4). The peer 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.

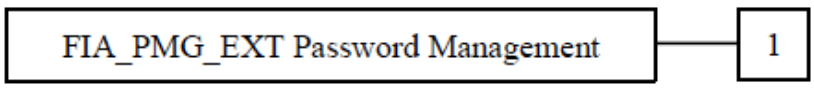
Identification and Authentication (FIA)

Password Management (FIA_PMG_EXT)

Family Behaviour

The TOE defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

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: FIA_PMG_EXT.1
No management functions.
Audit: FIA_PMG_EXT.1
No specific audit requirements.

FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1	Password Management
Hierarchical to:	No other components.
Dependencies:	No other components.
FIA_PMG_EXT.1.1	The TSF shall provide the following password management capabilities for administrative passwords: c) 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]]; d) Minimum password length shall be configurable to between [assignment: minimum number of characters supported by the TOE] and [assignment: number of characters greater than or equal to 15] characters.

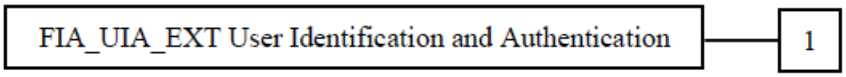
Application Note 203
The ST author selects the special characters that are supported by the TOE. They may optionally list additional special characters supported using the assignment. "Administrative passwords" refers to passwords used by Administrators at the local console, over protocols that support passwords, such as SSH and HTTPS, or to grant configuration data that supports other SFRs in the Security Target.
The second assignment should be configured with the largest minimum password length the Security Administrator can configure.

User Identification and Authentication (FIA_UIA_EXT)

Family Behaviour

The TSF allows certain specified actions before the non-TOE entity goes through the identification and authentication process.

Component levelling



FIA_UIA_EXT.1 User Identification and Authentication requires Administrators (including remote Administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions
Management: FIA_UIA_EXT.1
The following actions could be considered for the management functions in FMT:
a) Ability to configure the list of TOE services available before an entity is identified and authenticated
Audit: FIA_UIA_EXT.N
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) All use of the identification and authentication mechanism

b) Provided user identity, origin of the attempt (e.g. IP address)

FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1	User Identification and Authentication
Hierarchical to:	No other components.
Dependencies:	FTA_TAB.1 Default TOE Access Banners
FIA_UIA_EXT.1.1	The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process: • Display the warning banner in accordance with FTA_TAB.1; • [selection: no other actions, automated generation of cryptographic keys, [assignment: list of services, actions performed by the TSF in response to non-TOE requests]].
FIA_UIA_EXT.1.2	The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

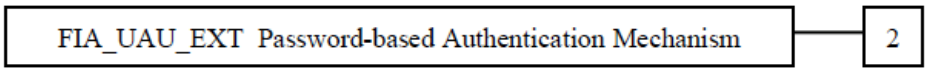
Application Note 204
This requirement applies to users (Administrators and external IT entities) of services available from the TOE directly, and not services available by connecting through the TOE. While it should be the case that few or no services are available to external entities prior to identification and authentication, if there are some available (perhaps ICMP echo) these should be listed in the assignment statement; if automated generation of cryptographic keys is supported without administrator authentication, the option "automated generation of cryptographic keys" should be selected; otherwise “no other actions” should be selected.
Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or be certificate based (such as SSH, TLS).
For communications with external IT entities (an audit server, for instance), such connections must be performed in accordance with FTP_ITC.1, whose protocols perform identification and authentication. This means that such communications (e.g., establishing the IPsec connection to the authentication server) would not have to be specified in the assignment, since establishing the connection “counts” as initiating the identification and authentication process.
According to the application note for FMT_SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 but not necessarily all TOE components. In case not all TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.

User authentication (FIA_UAU_EXT)

Family Behaviour

Provides for a locally based administrative user authentication mechanism

Component levelling



FIA_UAU_EXT.2 The password-based authentication mechanism provides administrative users a locally based authentication mechanism.
Management: FIA_UAU_EXT.2
The following actions could be considered for the management functions in FMT:
a) None
Audit: FIA_UAU_EXT.2
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) Minimal: All use of the authentication mechanism

FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2	**Password-based Authentication Mechanism**
Hierarchical to:	No other components.
Dependencies:	No other components.
FIA_UAU_EXT.2.1	The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: other authentication mechanism(s)], no other authentication mechanism] to perform local administrative user authentication.

Application Note 205
The assignment should be used to identify any additional local authentication mechanisms supported. Local authentication mechanisms are defined as those that occur through the local console; remote administrative sessions (and their associated authentication mechanisms) are specified in FTP_TRP.1/Admin.
According to the application note for FMT_SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 but not necessarily all TOE components. In case not all TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.

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:
a) Remove imported X.509v3 certificates
b) Approve import and removal of X.509v3 certificates
c) 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:
a) Minimal: No specific audit requirements are specified.

FIA_X509_EXT.1 X.509 Certificate Validation

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 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certification path validation.
- 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, no revocation method]
- The TSF shall validate the extendedKeyUsage field according to the following rules: [assignment: rules that govern contents of the extendedKeyUsage field that need to be verified].

Application Note 206

FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author selects whether revocation status is verified using OCSP or CRLs. If the TOE is distributed and X.509 based authentication is being used to authenticate the protocol selected in FPT_ITT.1, certificate revocation checking is optional. It is optional because there are additional requirements surrounding the enabling and disabling of the FPT_ITT channel defined in FCO_CPC_EXT.1. If revocation is not supported the ST author selects no revocation method. The ST author fills in the assignment with rules that may apply to other requirements in the ST. For instance, if a protocol such as TLS that uses certificates is specified in the ST, then certain values for the extendedKeyUsage field (e.g., “Server Authentication Purpose”) could be specified.

FIA_X509_EXT.1.2 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 207
This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

FIA_X509_EXT.2 X509 Certificate Authentication

FIA_X509_EXT.2 X.509 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 software updates, code signing for integrity verification, [assignment: other uses], no additional uses].

Application Note 208
If the TOE specifies the implementation of communications protocols that perform peer authentication using certificates, the ST author either selects or assigns the protocols that are specified; otherwise, they select “no protocols”. Protocols that do not use X.509 based peer authentication include SSH, where ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and/or ecdsa-sha2-nistp521 are selected. The TOE may also use certificates for other purposes; the second selection and assignment are used to specify these cases.

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 209
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. 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 valid according to all other rules in FIA_X509_EXT.1, 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. If the Administrator-configured option is selected by the ST Author, the ST Author also selects the corresponding function in FMT_SMF.1.
If the TOE is distributed and FIA_X509_EXT.1/ITT is selected, then certificate revocation checking is optional. This is due to additional authorization actions being performed in the

enabling and disabling of the intra-TOE trusted channel as defined in FCO_CPC_EXT.1. In this case, a connection is not required to determine certificate validity and this SFR is trivially satisfied.

FIA_X509_EXT.3 X.509 Certificate Requests

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.

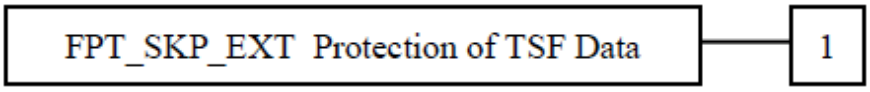
Protection of the TSF (FPT)

Protection of TSF Data (FPT_SKP_EXT)

Family Behaviour

Components in this family address the requirements for managing and protecting TSF data, such as cryptographic keys. This is a new family modelled after the FPT_PTD Class.

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: FPT_SKP_EXT.1
The following actions could be considered for the management functions in FMT:
a) There are no management activities foreseen.
Audit: FPT_SKP_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) There are no auditable events foreseen.

FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
Hierarchical to:	No other components.
Dependencies:	No other components.
FPT_SKP_EXT.1.1	The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

Application Note 210
The intent of this requirement is for the device to protect keys, key material, and authentication credentials from unauthorized disclosure. This data should only be accessed for the purposes of their assigned security functionality, and there is no need for them to be displayed/accessed at any other time. This requirement does not prevent the device from providing indication that these exist, are in use, or are still valid. It does, however, restrict the reading of the values outright.

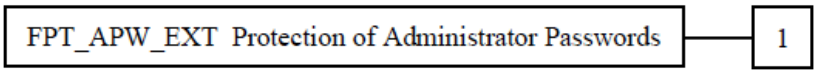
Protection of Administrator Passwords (FPT_APW_EXT)

FPT_APW_EXT.1 Protection of Administrator Passwords

Family Behaviour

Components in this family ensure that the TSF will protect plaintext credential data such as passwords from unauthorized disclosure.

Component levelling



FPT_APW_EXT.1 Protection of Administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.
Management: FPT_APW_EXT.1
The following actions could be considered for the management functions in FMT:
a) No management functions.
Audit: FPT_APW_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) No audit necessary.

FPT_APW_EXT.1	**Protection of Administrator Passwords**
Hierarchical to:	No other components
Dependencies:	No other components.
FPT_APW_EXT.1.1	The TSF shall store passwords in non-plaintext form.
FPT_APW_EXT.1.2	The TSF shall prevent the reading of plaintext passwords.

Application Note 211
The intent of the requirement is that raw password authentication data is not stored in the clear, and that no user or Administrator is able to read the plaintext password through “normal” interfaces. An all-powerful Administrator could directly read memory to capture a password but is trusted not to do so. Passwords should be obscured during entry on the local console in accordance with FIA_UAU.7.

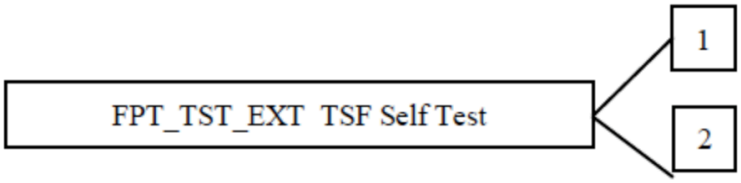
TSF Self-Test (FPT_TST_EXT)

FPT_TST_EXT.1 TSF Testing

Family Behaviour

Components in this family address the requirements for self-testing the TSF for selected correct operation.

Component levelling



FPT_TST_EXT.1 TSF Self-Test requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF.
FPT_TST_EXT.2 Self-tests based on certificates applies when using certificates as part of self-test, and requires that the self-test fails if a certificate is invalid.
Management: FPT_TST_EXT.1, FPT_TST_EXT.2
The following actions could be considered for the management functions in FMT:
a) No management functions.
Audit: FPT_TST_EXT.1, FPT_TST_EXT.2
The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:
a) Indication that TSF self-test was completed
b) Failure of self-test

FPT_TST_EXT.1	**TSF testing**
Hierarchical to:	No other components.
Dependencies:	No other components.
FPT_TST_EXT.1.1	The TSF shall run a suite of the following self-tests [selection: during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]] to demonstrate the correct operation of the TSF: [assignment: list of self-tests run by the TSF].

Application Note 212

It is expected that self-tests are carried out during initial start-up (on power on). Other options should only be used if the developer can justify why they are not carried out during initial start-up. It is expected that at least self-tests for verification of the integrity of the firmware and software as well as for the correct operation of cryptographic functions necessary to fulfil the SFRs will be performed. If not all self-tests are performed during start-up multiple iterations of this SFR are used with the appropriate options selected. In future versions of this cPP the suite of self-tests will be required to contain at least mechanisms for measured boot including self-tests of the components which perform the measurement.
Non-distributed TOEs may internally consist of several components that contribute to enforcing SFRs. Self-testing shall cover all components that contribute to enforcing SFRs and verification of integrity shall cover all software that contributes to enforcing SFRs on all components.
For distributed TOEs all TOE components have to perform self-tests. This does not necessarily mean that each TOE component has to carry out the same self-tests: the ST describes the applicability of the selection (i.e. when self-tests are run) and the final assignment (i.e. which self-tests are carried out) to each TOE component.

Application Note 213
If certificates are used by the self-test mechanism (e.g. for verification of signatures for integrity verification), certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TST_EXT.2 must be included in the ST.

FPT_TST_EXT.2 Self-tests based on certificates

Hierarchical to: No other components.

Dependencies: No other components.

FPT_TST_EXT.2.1 The TSF shall fail self-testing if a certificate is used for self-tests and the corresponding certificate is deemed invalid.

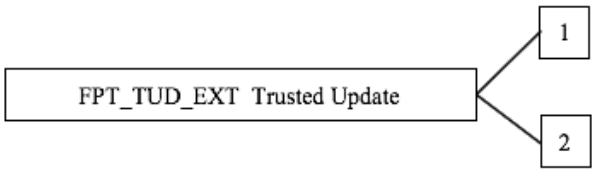
Application Note 214
Certificates may optionally be used for self-tests (FPT_TST_EXT.1.1). This element must be included in the ST if certificates are used for self-tests. If “code signing for integrity verification” is selected in FIA_X509_EXT.2.1, FPT_TST_EXT.2 must be included in the ST.
Validity is determined by the certification path and the expiration date. If the self-test is executed as part of TOE initialization (e.g. boot), there is no expectation of a revocation status check as the necessary functionality, configuration, or infrastructure required to perform such check might not be available.

Trusted Update (FPT_TUD_EXT)

Family Behaviour

Components in this family address the requirements for updating the TOE firmware and/or software.

Component levelling



FPT_TUD_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.
FPT_TUD_EXT.2 Trusted update based on certificates applies when using certificates as part of trusted update and requires that the update does not install if a certificate is invalid.
Management: FPT_TUD_EXT.1
The following actions could be considered for the management functions in FMT:
a) Ability to update the TOE and to verify the updates
b) Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1/SigGen) and [selection: no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]]
c) Ability to update the TOE, and to verify the updates using [selection: digital signature, published hash, no other mechanism] capability prior to installing those updates
Audit: FPT_TUD_EXT.1, FPT_TUD_EXT.2
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) Initiation of the update process.
b) Any failure to verify the integrity of the update

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1 Trusted update

Hierarchical to:	No other components
Dependencies:	FCS_COP.1/SigGen Cryptographic operation (for Cryptographic Signature and Verification), or FCS_COP.1/Hash Cryptographic operation (for cryptographic hashing)
FPT_TUD_EXT.1.1	The TSF shall provide [assignment: Administrators] the ability to query the currently executing version of the TOE firmware/software and [selection: the most recently installed version of the TOE firmware/software; no other TOE firmware/software version].

Application Note 215
If a trusted update can be installed on the TOE with a delayed activation the version of both the currently executing image and the installed but inactive image must be provided. In this case the option 'the most recently installed version of the TOE firmware/software' needs to be chosen from the selection in FPT_TUD_EXT.1.1 and the TSS needs to describe how and when the inactive version becomes active. If all trusted updates become active as part of the installation process, only the currently executing version needs to be provided. In this case the option 'no other TOE firmware/software version' shall be chosen from the selection in FPT_TUD_EXT.1.1..
For a distributed TOE, the method of determining the installed versions on each component of the TOE is described in the operational guidance.

FPT_TUD_EXT.1.2	The TSF shall provide [assignment: Administrators] the ability to manually initiate updates to TOE firmware/software and [selection:

support automatic checking for updates, support automatic updates, no other update mechanism].

Application Note 216
The selection in FPT_TUD_EXT.1.2 distinguishes the support of automatic checking for updates and support of automatic updates. The first option refers to a TOE that checks whether a new update is available, communicates this to the Administrator (e.g. through a message during an Administrator session, through log files) but requires some action by the Administrator to actually perform the update. The second option refers to a TOE that checks for updates and automatically installs them upon availability.
The TSS explains what actions are involved in the TOE support when using the “support automatic checking for updates” or “support automatic updates” selections.
When published hash values (see FPT_TUD_EXT.1.3) are used to protect the trusted update mechanism, the TOE must not automatically download the update file(s) together with the hash value (either integrated in the update file(s) or separately) and automatically install the update without any active authorization by the Security Administrator, even when the calculated hash value matches the published hash value. When using published hash values to protect the trusted update mechanism, the option “support of automatic updates” must not be used (automated checking for updates is permitted, though). The TOE may automatically download the update file(s) themselves but not to the hash value. For the published hash approach, it is intended that a Security Administrator is always required to give active authorisation for installation of an update (as described in more detail under FPT_TUD_EXT.1.3) below. Due to this, the type of update mechanism is regarded as “manually initiated update”, even if the update file(s) may be downloaded automatically. A fully automated approach (without Security Administrator intervention) can only be used when “digital signature mechanism” is selected in FPT_TUD_EXT.1.3 below.

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates.

Application Note 217
The digital signature mechanism referenced in the selection of FPT_TUD_EXT.1.3 is one of the algorithms specified in FCS_COP.1/SigGen. The published hash referenced in FPT_TUD_EXT.1.3 is generated by one of the functions specified in FCS_COP.1/Hash. The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.
When published hash values are used to secure the trusted update mechanism, an active authorization of the update process by the Security Administrator is always required. The secure transmission of an authentic hash value from the developer to the Security Administrator is one of the key factors to protect the trusted update mechanism when using published hashes and the guidance documentation needs to describe how this transfer has to be performed. For the verification of the trusted hash value by the Security Administrator different use cases are possible. The Security Administrator could obtain the published hash value as well as the update file(s) and perform the verification outside the TOE while the hashing of the update file(s) could be done by the TOE or by other means. Authentication as Security Administrator and initiation of the trusted update would in this case be regarded as “active authorization” of the trusted update. Alternatively, the Administrator could provide the TOE with the published hash value together with the update file(s) and the hashing and hash comparison is performed by the TOE. In case of successful hash verification, the TOE can perform the update without any additional step by the Security Administrator. Authentication as Security Administrator and sending the hash value to the TOE is regarded as “active authorization” of the trusted update (in case of successful hash verification), because the Administrator is expected to load the hash value only to the TOE when intending to perform the update. As long as the transfer of the hash value to the TOE is performed by the Security Administrator, loading of the update file(s) can be performed by the Security Administrator or can be automatically downloaded by the TOE from a repository.

If the digital signature mechanism is selected, the verification of the signature shall be performed by the TOE itself. For the published hash option, the verification can be done by the TOE itself as well as by the Security Administrator. In the latter case use of TOE functionality for the verification is not mandated, so verification could be done using non-TOE functionality of the device containing the TOE or without using the device containing the TOE.
For distributed TOEs all TOE components shall support Trusted Update. The verification of the signature or hash on the update shall either be done by each TOE component itself (signature verification) or for each component (hash verification).
Updating a distributed TOE might lead to the situation where different TOE components are running different software versions. Depending on the differences between the different software versions the impact of a mixture of different software versions might be no problem at all or critical to the proper functioning of the TOE. The TSS shall detail the mechanisms that support the continuous proper functioning of the TOE during trusted update of distributed TOEs.

Application Note 218
Future versions of this cPP will mandate the use of a digital signature mechanism for trusted updates.
Application Note 219
If certificates are used by the update verification mechanism, certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TUD_EXT.2 must be included in the ST.
Application Note 220
“Update” in the context of this SFR refers to the process of replacing a non-volatile, system resident software component with another. The former is referred to as the NV image, and the latter is the update image. While the update image is typically newer than the NV image, this is not a requirement. There are legitimate cases where the system owner may want to rollback a component to an older version (e.g. when the component manufacturer releases a faulty update, or when the system relies on an undocumented feature no longer present in the update). Likewise, the owner may want to update with the same version as the NV image to recover from faulty storage.
All discrete firmware and software elements (e.g. applications, drivers, and kernel) of the TSF need to be protected, i.e. they should either be digitally signed by the corresponding manufacturer and subsequently verified by the mechanism performing the update or a hash should be published for them which needs to be verified before the update.

FPT_TUD_EXT.2 Trusted Update based on certificates

FPT_TUD_EXT.2 Trusted update based on certificates

Hierarchical to: No other components

Dependencies: FPT_TUD_EXT.1

FPT_TUD_EXT.2.1 The TSF shall not install an update if the code signing certificate is deemed invalid.

FPT_TUD_EXT.2.2 When the certificate is deemed invalid because the certificate has expired, 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 221
Certificates may optionally be used for code signing of system software updates (FPT_TUD_EXT.1.3). This element must be included in the ST if certificates are used for

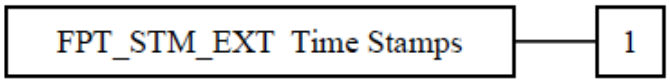
validating updates. If “code signing for system software updates” is selected in FIA_X509_EXT.2.1, FPT_TUD_EXT.2 must be included in the ST.
Validity is determined by the certification path, the expiration date, and the revocation status in accordance with FIA_X509_EXT.1. For expired certificates the author of the ST selects whether the certificate shall be accepted, rejected or the choice is left to the Administrator to accept or reject the certificate.

Time stamps (FPT_STM_EXT)

Family Behaviour

Components in this family extend FPT_STM requirements by describing the source of time used in timestamps.

Component levelling



FPT_STM_EXT.1 Reliable Time Stamps is hierarchic to FPT_STM.1: it requires that the TSF provide reliable time stamps for TSF and identifies the source of the time used in those timestamps.
Management: FPT_STM_EXT.1
The following actions could be considered for the management functions in FMT:
a) Management of the time
b) Administrator setting of the time.
Audit: FTA_SSL_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) Discontinuous changes to the time.

FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1	**Reliable Time Stamps**
Hierarchical to:	No other components
Dependencies:	No other components.
FPT_STM_EXT.1.1	The TSF shall be able to provide reliable time stamps for its own use.
FPT_STM_EXT.1.2	The TSF shall [selection: allow the Security Administrator to set the time, synchronise time with an NTP server].

Application Note 222
Reliable time stamps are expected to be used with other TSF, e.g. for the generation of audit data to allow the Security Administrator to investigate incidents by checking the order of events and to determine the actual local time when events occurred. The decision about the required level of accuracy of that information is up to the Administrator.
The TOE depends on time and date information, either provided by a local real-time clock that is manually managed by the Security Administrator or through the use of one or more external NTP

servers. The corresponding option(s) shall be chosen from the selection in FPT_STM_EXT.1.2. The use the automatic synchronisation with an external NTP server is recommended but not mandated. Note that for the communication with an external NTP server, FCS_NTP_EXT.1 shall be claimed. The ST author describes in the TSS how the external time and date information is received by the TOE and how this information is maintained.
The term “reliable time stamps” refers to the strict use of the time and date information, that is provided, and the logging of all discontinuous changes to the time settings including information about the old and new time. With this information, the real time for all audit data can be determined. Note, that all discontinuous time changes, Administrator actuated or changed via an automated process, must be audited. No audit is needed when time is changed via use of kernel or system facilities – such as daytime (3) – that exhibit no discontinuities in time.
For distributed TOEs it is expected that the Security Administrator ensures synchronization between the time settings of different TOE components. All TOE components shall either be in sync (e.g. through synchronisation between TOE components or through synchronisation of different TOE components with external NTP servers) or the offset should be known to the Administrator for every pair of TOE components. This includes TOE components synchronized to different time zones.

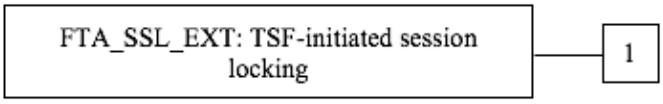
TOE Access (FTA)

TSF-initiated Session Locking (FTA_SSL_EXT)

Family Behaviour

Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.
The extended FTA_SSL_EXT family is based on the FTA_SSL family.

Component levelling



FTA_SSL_EXT.1 TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.
Management: FTA_SSL_EXT.1
The following actions could be considered for the management functions in FMT:
c) Specification of the time of user inactivity after which lock-out occurs for an individual user.
Audit: FTA_SSL_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
b) Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1 TSF-initiated Session Locking

Hierarchical to:	No other components
Dependencies:	FIA_UAU.1 Timing of authentication
FTA_SSL_EXT.1.1	The TSF shall, for local interactive sessions, [selection: • lock the session - disable any activity of the Administrator's data access/display devices other than unlocking the session, and requiring that the Administrator re-authenticate to the TSF prior to unlocking the session; • terminate the session] after a Security Administrator-specified time period of inactivity.

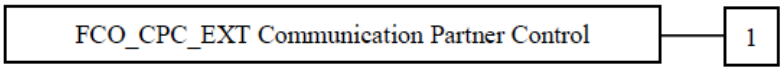
Communication (FCO)

Communication Partner Control (FCO_CPC_EXT)

Family Behaviour

This family is used to define high-level constraints on the ways that partner IT entities communicate. For example, there may be constraints on when communication channels can be used, how they are established, and links to SFRs expressing lower-level security properties of the channels.

Component levelling



FCO_CPC_EXT.1 Component Registration Channel Definition, requires the TSF to support a registration channel for joining together components of a distributed TOE, and to ensure that the availability of this channel is under the control of an Administrator. It also requires statement of the type of channel used (allowing specification of further lower-level security requirements by reference to other SFRs).
Management: FCO_CPC_EXT.1
No separate management functions are required. Note that elements of the SFR already specify certain constraints on communication in order to ensure that the process of forming a distributed TOE is a controlled activity.
Audit: FCO_CPC_EXT.1
The following actions should be auditable if FCO_CPC_EXT.1 is included in the PP/ST:
a) Enabling communications between a pair of components as in FCO_CPC_EXT.1.1 (including identities of the endpoints).
b) Disabling communications between a pair of components as in FCO_CPC_EXT.1.3 (including identity of the endpoint that is disabled).
If the required types of channel in FCO_CPC_EXT.1.2 are specified by using other SFRs then the use of the registration channel may be sufficiently covered by the audit requirements on those SFRs: otherwise a separate audit requirement to audit the use of the channel should be identified for FCO_CPC_EXT.1.

FCO_CPC_EXT.1 Component Registration Channel Definition

FCO_CPC_EXT.1 Component Registration Channel Definition

Hierarchical to: No other components.
Dependencies: No other components.

FCO_CPC_EXT.1.1 The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

FCO_CPC_EXT.1.2 The TSF shall implement a registration process in which components establish and use a communications channel that uses [assignment: list of different types of channel given in the form of a selection] for at least [assignment: type of data for which the channel must be used].

FCO_CPC_EXT.1.3 The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.

Application Note 223
This SFR is generally applied to a distributed TOE in order to control the process of creating the distributed TOE from its components by means of a registration process in which a component joins the distributed TOE by registering with an existing component of the distributed TOE. When creating the TSF from the initial pair of components, either of these components may be identified as the TSF for the purposes of satisfying the meaning of “TSF” in this SFR.
The intention of this requirement is to ensure that there is a registration process that includes a positive enablement step by an Administrator before components joining a distributed TOE can communicate with the other components of the TOE and before the new component can act as part of the TSF. The registration process may itself involve communication with the joining component: many network devices use a bespoke process for this, and the security requirements for the “registration communication” are then defined in FCO_CPC_EXT.1.2. Use of this “registration communication” channel is not deemed inconsistent with the requirement of FCO_CPC_EXT.1.1 (i.e. the registration channel can be used before the enablement step, but only in order to complete the registration process).