

PP-Module for LiFi Access Systems



Version: 1.0
2026-02-24

National Information Assurance Partnership

Revision History

Version	Date	Comment
1.0	2026-02-24	Initial Release

Contents

1	Introduction
1.1	Overview
1.2	Terms
1.2.1	Common Criteria Terms
1.2.2	Technical Terms
1.3	Compliant Targets of Evaluation
1.3.1	TOE Boundary
1.4	Use-Cases
1.5	Implementation-Based Features
1.5.1	802.11bb Support
1.5.2	Distributed TOE
1.5.3	G.vlc Support
2	Conformance Claims
3	Security Problem Definition
3.1	Threats
3.2	Assumptions
3.3	Organizational Security Policies
4	Security Objectives
4.1	Security Objectives for the Operational Environment
4.2	Security Objectives Rationale
5	Security Requirements
5.1	Collaborative Protection Profile for Network Device Security Functional Requirements Direction
5.1.1	Modified SFRs
5.2	TOE Security Functional Requirements
5.2.1	Auditable Events for Mandatory SFRs
5.2.2	Security Audit (FAU)
5.2.3	Cryptographic Support (FCS)
5.2.4	Identification and Authentication (FIA)
5.2.5	Security Management (FMT)
5.2.6	Protection of the TSF (FPT)
5.2.7	TOE Access (FTA)
5.2.8	Trusted Path/Channels (FTP)
5.3	TOE Security Functional Requirements Rationale
6	Consistency Rationale
6.1	Collaborative Protection Profile for Network Device
6.1.1	Consistency of TOE Type
6.1.2	Consistency of Security Problem Definition
6.1.3	Consistency of OE Objectives
6.1.4	Consistency of Requirements

Appendix A - Optional SFRs

A.1 Strictly Optional Requirements

A.2 Objective Requirements

A.3 Implementation-dependent Requirements

A.3.1 Auditable Events for Implementation-Dependent SFRs

A.3.2 SFRs for 802.11bb Implementations

A.3.3 SFRs for Distributed TOE Implementations

A.3.4 SFRs for G.vlc Implementations

Appendix B - Selection-based Requirements

B.1 Auditable Events for Selection-based SFRs

B.2 Cryptographic Support (FCS)

Appendix C - Extended Component Definitions

C.1 Extended Components Table

C.2 Extended Component Definitions

C.2.1 Cryptographic Support (FCS)

C.2.1.1 FCS_RADSEC_EXT RadSec

C.2.2 Identification and Authentication (FIA)

C.2.2.1 FIA_8021X_EXT 802.1X Port Access Entity (Authenticator) Authentication

C.2.2.2 FIA_PSK_EXT Pre-Shared Keys

C.2.3 Protection of the TSF (FPT)

C.2.3.1 FPT_ACF_EXT Access Controls

C.2.3.2 FPT_ASLR_EXT Address Space Layout Randomization

C.2.3.3 FPT_SBOP_EXT Stack Buffer Overflow Protection

C.2.3.4 FPT_W^X_EXT Write XOR Execute Memory Pages

C.2.4 Security Management (FMT)

C.2.4.1 FMT_SMR_EXT Security Management Restrictions

Appendix D - Acronyms

Appendix E - Bibliography

1 Introduction

1.1 Overview

The scope of the LiFi Access System PP-Module is to describe the security functionality of a LiFi access system product in terms of [CC] and to define functional and assurance requirements for such products. This PP-Module is intended for use with the following Base-PPs:

- collaborative Protection Profile for Network Devices, Version 4.0

A conformant TOE will claim conformance to a PP-Configuration that includes the Base-PP identified above, this PP-Module, and any of the PP-Modules or packages identified in Section 2.

This Base-PP is valid because a LiFi Access System is a specific function for a purpose-built network device.

1.2 Terms

The following sections list Common Criteria and technology terms used in this document.

1.2.1 Common Criteria Terms

Assurance	Grounds for confidence that a TOE meets the SFRs [CC].
Base Protection Profile (Base-PP)	Protection Profile used as a basis to build a PP-Configuration.
Collaborative Protection Profile (cPP)	A Protection Profile developed by international technical communities and approved by multiple schemes.
Common Criteria (CC)	Common Criteria for Information Technology Security Evaluation (International Standard ISO/IEC 15408).
Common Criteria Testing Laboratory	Within the context of the Common Criteria Evaluation and Validation Scheme (CCEVS), an IT security evaluation facility accredited by the National Voluntary Laboratory Accreditation Program (NVLAP) and approved by the NIAP Validation Body to conduct Common Criteria-based evaluations.
Common Evaluation Methodology (CEM)	Common Evaluation Methodology for Information Technology Security Evaluation.
Distributed TOE	A TOE composed of multiple components operating as a logical whole.
Operational Environment (OE)	Hardware and software that are outside the TOE boundary that support the TOE functionality and security policy.
Protection Profile (PP)	An implementation-independent set of security requirements for a category of products.

Protection Profile Configuration (PP-Configuration)	A comprehensive set of security requirements for a product type that consists of at least one Base-PP and at least one PP-Module.
Protection Profile Module (PP-Module)	An implementation-independent statement of security needs for a TOE type complementary to one or more Base-PPs.
Security Assurance Requirement (SAR)	A requirement to assure the security of the TOE.
Security Functional Requirement (SFR)	A requirement for security enforcement by the TOE.
Security Target (ST)	A set of implementation-dependent security requirements for a specific product.
Target of Evaluation (TOE)	The product under evaluation.
TOE Security Functionality (TSF)	The security functionality of the product under evaluation.
TOE Summary Specification (TSS)	A description of how a TOE satisfies the SFRs in an ST.

1.2.2 Technical Terms

Access Point (AP)	A device that provides the network interface that enables wireless client hosts to access a wired network.
LiFi	A wireless computer network that links two or more devices using modulated optical communication to form a local area network within a limited area such as a home, school, computer laboratory, campus, office building, etc.
Service Set Identifier (SSID)	The primary name associated with an 802.11 wireless local area network (WLAN).

1.3 Compliant Targets of Evaluation

This ~~PP-Module~~ specifically addresses LiFi (IEEE 802.11bb, ITU G.9991) access systems (AS). A compliant LiFi AS is a system composed of hardware and software that is connected to a network and has an infrastructure role in the overall enterprise network. In particular, a LiFi AS establishes a secure wireless (IEEE 802.11, IEEE802.1x, IEEE 802.1AE) link that provides an authenticated and encrypted path to an enterprise network and thereby decreases the risk of exposure of information transiting “over-the-air.”

Since this ~~PP-Module~~ extends the ~~NDcPP~~, conformant ~~TOEs~~ are obligated to implement the functionality required in the ~~NDcPP~~ along with the additional functionality defined in this ~~PP-Module~~ in response to the threat environment discussed subsequently herein.

The ~~TOE~~ encompasses the OS kernel drivers, shared software libraries, and some application software included with the OS. The applications considered within the ~~TOE~~ are those that provide essential LiFi services that make up the LiFi link and security. Examples are management services that implement configuration functions specified in this ~~PP-Module~~, host services that manage client authentication to the access point, and the libraries and modules

that provide security implementation of the LiFi link. This applies to each separate **TOE** component that is being used to meet the security requirements proposed.

1.3.1 TOE Boundary

The physical boundary for a **TOE** that conforms to this **PP-Module** is one or more physical network appliances (in a standalone or distributed configuration) that provides generalized network device functionality such as auditing, identification and authentication, and cryptographic services for network communications, as well as specialized functionality for facilitating remote network access via LiFi communications. The **TOE**'s logical boundary includes all functionality required by the claimed **Base-PP**, this **PP-Module**, and any other **PP-Modules** that may be claimed by the **ST** author as a part of a **PP-Configuration** that includes this **PP-Module**. Product functionality for which no **PP-Module** exists, or for which no **SFR** within a relevant **PP-Module** exists to capture, are considered to be non-interfering with respect to the **TSF**. In other words, they may be present or enabled in the evaluated configuration of the **TOE** but their presence must not degrade or interrupt the enforcement of functions within the **TOE**'s logical boundary.

Note that while the **Base-PP** permits the **TOE** to be either a physical or virtual network appliance, LiFi devices require the use of specialized hardware that will not be present on commodity servers, and it is therefore expected that any **TOE** that conforms to this **PP-Module** will be one or more dedicated physical devices.

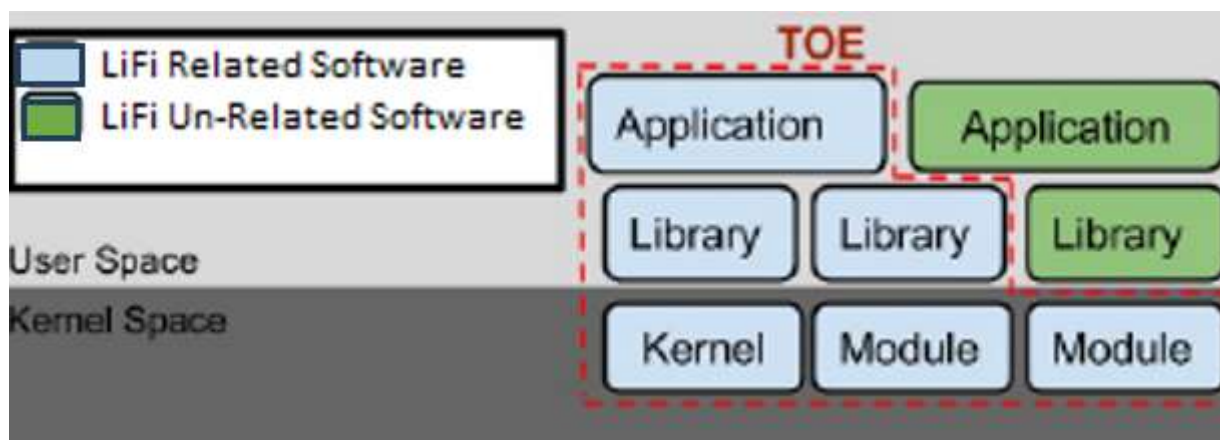


Figure 1: High Level **TOE** Implementation

1.4 Use-Cases

[USE CASE 1] Standalone Device

The **TOE** is a standalone network device that serves as a single network endpoint providing connectivity to LiFi clients.

[USE CASE 2] Distributed System

The **TOE** is a distributed system consisting of multiple network devices that collectively serve as the LiFi network endpoint. In addition to claiming the relevant "Distributed **TOE**" requirement in the **NDcPP**, this use case also requires the **TOE** to claim the optional **SFR** [FCS_CKM.2/DISTRIB](#) to describe the key distribution method between distributed **TOE** components.

1.5 Implementation-Based Features

A conformant **TOE** may implement either 802.11bb or G.vlc. Depending on which is supported, different requirements will apply to the **TOE** as they each implement different methods of protection of data in transit.

1.5.1 802.11bb Support

The TOE implements LiFi using IEEE 802.11bb. Data in transit is protected using either IPsec or RadSec (RADIUS over TLS).

If this feature is implemented by the TOE, the following requirements must be claimed in the ST:

- [FCS_CKM.1/WPA](#)
- [FCS_CKM.2/GTK](#)
- [FCS_CKM.2/PMK](#)
- [FTP_ITC.1/Client](#)

1.5.2 Distributed TOE

A specific subtype of a TOE that supports 802.11bb is a distributed architecture. When the TOE implements this feature, it uses a key distribution mechanism to pass keys between distributed components so that a remote client may persistently maintain a connection to different connections in transit (e.g., when moving to different floors of a building).

If this feature is implemented by the TOE, the following requirements must be claimed in the ST:

- [FCS_CKM.2/DISTRIB](#)

1.5.3 G.vlc Support

The TSF implements LiFi using ITU-T G.9991 (aka G.vlc). Data in transit is protected using MACsec. When the TOE implements this feature, it must also claim conformance to the PP-Module for MACsec Ethernet Encryption.

If this feature is implemented by the TOE, the following requirements must be claimed in the ST:

- [FTP_ITC.1/Gvlc](#)

2 Conformance Claims

Conformance Statement

An ST must claim exact conformance to this PP-Module.

The evaluation methods used for evaluating the TOE are a combination of the workunits defined in [\[CEM\]](#) as well as the Evaluation Activities for ensuring that individual SFRs and SARs have a sufficient level of supporting evidence in the Security Target and guidance documentation and have been sufficiently tested by the laboratory as part of completing ATE_IND.1. Any functional packages this PP claims similarly contain their own Evaluation Activities that are used in this same manner.

CC Conformance Claims

This PP-Module is conformant to Part 2 (extended) and Part 3 (conformant) of Common Criteria CC:2022, Revision 1.

PP Claim

This PP-Module does not claim conformance to any Protection Profile.

The following PPs and PP-Modules are allowed to be specified in a PP-Configuration with this PP-Module:

- collaborative Protection Profile for Network Devices, Version 4.0
- PP-Module for MACsec Ethernet Encryption, version 2.0

Package Claim

- This PP-Module is Functional Package for Transport Layer Security Version 2.1 conformant.
- This PP-Module is Functional Package for X.509 Version 1.0 conformant.
- This PP-Module does not conform to any assurance packages.

The functional packages to which the PP conforms may include SFRs that are not mandatory to claim for the sake of conformance. An ST that claims one or more of these functional packages may include any non-mandatory SFRs that are appropriate to claim based on the capabilities of the TSF and on any triggers for their inclusion based inherently on the SFR selections made.

3 Security Problem Definition

This PP-Module is written to address the situation where a distributed network device facilitates end user access to a network via LiFi. The threats are similar to those that are typical of network devices as a whole, but may be exploited differently because of the mechanisms that are unique to this particular type of product.

3.1 Threats

T.DATA_INTEGRITY

Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to modify the data without authorization. If known malicious external devices are able to communicate with devices on the protected network or if devices on the protected network can establish communications with those external devices, then the data contained within the communications may be susceptible to a loss of integrity.

T.NETWORK_DISCLOSURE

Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to conduct unauthorized activities. If malicious external devices are able to communicate with devices on the protected network, or if devices on the protected network can establish communications with those external devices (e.g., as a result of nonexistent or insufficient data encryption that exposes the data in transit to rogue elements), then those internal devices may be susceptible to the unauthorized disclosure of information.

T.NETWORK_ACCESS

Devices located outside the protected network may seek to exercise services located on the protected network that are intended to be only accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network.

T.NETWORK_ATTACK

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with applications and services running on or part of the T.OE with the intent of compromise. Engagement may consist of altering existing legitimate communications.

T.REPLAY_ATTACK

If an unauthorized individual successfully gains access to the system, the adversary may have the opportunity to conduct a "replay" attack. This method of attack allows the individual to capture packets traversing throughout the wireless network and send the packets later, possibly unknown by the intended receiver.

3.2 Assumptions

These assumptions are made on the Operational Environment (OE) in order to be able to ensure that the security functionality specified in the PP-Module can be provided by the T.OE. If the T.OE is placed in an OE that does not meet these assumptions, the T.OE may no longer be able to provide all of its security functionality.

A.NON_SECURITY_FUNCTIONS

The product implements some security-relevant functionality that does not require evaluation (e.g., network time synchronization, process scheduling, and virtual memory management including process separation).

A.MANAGED_CONFIG

Depending on configuration and capability, the product may or may not be configuration-managed by the enterprise or bound to directory services to support multi-user login.

A.THIRD_PARTY_SOFTWARE

The product runs application software developed by a third party. The applications are not intentionally developed to be malicious, but can contain inadvertent coding errors. These errors introduce a risk that control of an application may be seized by a malicious entity. The product shall confine these applications within the originally designated operating environment.

A.NETWORK_CONNECTIVITY

The platform is connected to a network. For purposes of sending and receiving data, to include software updates, the platform is connected to other entities. Other entities on the network are not inherently trustable.

A.PROPER_USER

Users are not malicious in nature, though they may inadvertently or intentionally engage in risky behavior.

3.3 Organizational Security Policies

This document does not define any additional OSPs.

4 Security Objectives

4.1 Security Objectives for the Operational Environment

The OE of the TOE implements technical and procedural measures to assist the TOE in correctly providing its security functionality (which is defined by the security objectives for the TOE). The security objectives for the OE consist of a set of statements describing the goals that the OE should achieve. This section defines the security objectives that are to be addressed by the IT domain or by non-technical or procedural means. The assumptions identified in Section 3 are incorporated as security objectives for the environment.

OE.NON_SECURITY_FUNCTIONS

The product has functionality outside of its logical security boundary to satisfy organizational requirements that are unrelated to what this PP-Module defines.

OE.MANAGED_CONFIG

The product is compatible with enterprise architecture that allows for managed configuration, subject authorization via directory services, or both.

OE.THIRD_PARTY_SOFTWARE

The operational environment permits the use of third-party software on organization-owned hardware to enforce necessary functions.

OE.NETWORK_CONNECTIVITY

The operational environment facilitates the use of network connectivity by authorized subjects.

OE.PROPER_USER

Users of the system are adequately vetted to ensure that they are non-malicious.

4.2 Security Objectives Rationale

This section describes how the assumptions and organizational security policies map to operational environment security objectives.

Table 1: Security Objectives Rationale

Assumption or OSP	Security Objectives	Rationale
A.NON_SECURITY_FUNCTIONS	OE.NON_SECURITY_FUNCTIONS	The operational environment objective OE.NON_SECURITY_FUNCTIONS is realized through A.NON_SECURITY_FUNCTIONS.
A.MANAGED_CONFIG	OE.MANAGED_CONFIG	The operational environment objective OE.MANAGED_CONFIG is realized through A.MANAGED_CONFIG.
A.THIRD_PARTY_SOFTWARE	OE.THIRD_PARTY_SOFTWARE	The operational environment objective OE.THIRD_PARTY_SOFTWARE is realized through A.THIRD_PARTY_SOFTWARE.

A.NETWORK_CONNECTIVITY	OE.NETWORK_CONNECTIVITY	The operational environment objective OE.NETWORK_CONNECTIVITY is realized through A.NETWORK_CONNECTIVITY .
A.PROPER_USER	OE.PROPER_USER	The operational environment objective OE.PROPER_USER is realized through A.PROPER_USER .

5 Security Requirements

This chapter describes the security requirements which have to be fulfilled by the product under evaluation. Those requirements comprise functional components from Part 2 and assurance components from Part 3 of [CC]. The following conventions are used for the completion of operations:

- **Refinement** operation (denoted by **bold text** or ~~striketrough-text~~): Is used to add details to a requirement or to remove part of the requirement that is made irrelevant through the completion of another operation, and thus further restricts a requirement.
- **Selection** (denoted by *italicized text*): Is used to select one or more options provided by the [CC] in stating a requirement.
- **Assignment** operation (denoted by *italicized text*): Is used to assign a specific value to an unspecified parameter, such as the length of a password. Showing the value in square brackets indicates assignment.
- **Iteration** operation: Is indicated by appending the SFR name with a slash and unique identifier suggesting the purpose of the operation, e.g. "/EXAMPLE1."

5.1 Collaborative Protection Profile for Network Device Security Functional Requirements Direction

In a ~~PP-Configuration~~ that includes the NDcPP, the TOE is expected to rely on some of the security functions implemented by the network device as a whole and evaluated against the Base-PP. In this case, the following sections describe any modifications that the ST author must make to the SFRs defined in the Base-PP in addition to what is mandated by section 5.2.

5.1.1 Modified SFRs

This ~~PP-Module~~ does not modify any SFRs defined by the NDcPP.

5.2 TOE Security Functional Requirements

The following section describes the SFRs that must be satisfied by any TOE that claims conformance to this PP-Module. These SFRs must be claimed regardless of which PP-Configuration is used to define the TOE.

5.2.1 Auditable Events for Mandatory SFRs

Table 2: Auditable Events for Mandatory Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1/LiFi	No events specified	N/A
FCS_COP.1/LiFiDataEncryption	No events specified	N/A
FIA_8021X_EXT.1	None	Provided client identity (e.g., MAC address)

FIA_UAU.6	None	Origin of the attempt (e.g., IP address)
FMT_SMF.1/LiFi	No events specified	N/A
FMT_SMR_EXT.1	No events specified	N/A
FPT_ACF_EXT.1	Unauthorized attempts to perform operations against protected data	No additional information
FPT_ASLR_EXT.1	No events specified	N/A
FPT_SBOP_EXT.1	No events specified	N/A
FPT_W^X_EXT.1	No events specified	N/A
FTA_TSE.1	Denial of a session establishment due to the session establishment mechanism	- Reason for denial - Origin of establishment attempt
FTP_ITC.1/8021X	Failed attempts to establish a trusted channel (including IEEE 802.11)	Identification of the initiator and target of channel
	Detection of modification of channel data	Identification of the initiator and target of channel

5.2.2 Security Audit (FAU)

FAU_GEN.1/LiFi Audit Data Generation (LiFi)

FAU_GEN.1.1/LiFi

The T_{SSF} shall be able to generate audit data of the following auditable events:

- Start-up and shutdown of the audit functions
- All auditable events for the [*not specified*] level of audit
- [*auditable events defined in Table 2*]
- [*failure of wireless sensor communication*]
- [**selection:** *auditable events defined in Table 8, no additional auditable events*]

]

Application Note: A T_{OE} that conforms to a Protection Profile Configuration (PP-Configuration) containing this PP-Module can either be a standalone or distributed T_{OE} as defined in the NDcPP. For distributed T_{OE}s, the expectation for this PP-Module is that a LiFi AS is composed of a single controller and one or more access points (APs). If the T_{OE} is a distributed system across multiple components, then they must be capable of sending audit data to the controller. If the T_{OE} is a standalone system, audit data should be stored on the component that generates it.

The auditable events defined in Table 2, Table 8, and Table 10 are for the S_{FR}s that are explicitly defined in this PP-Module and are intended to extend FAU_GEN.1 in the Base-PP (Table 10 is not referenced in the S_{FR} as none of the S_{FR}s in that section have auditable events associated with them). The events in the Auditable

Events table should be combined with those of the NDcPP in the context of a conforming Security Target. This table includes rows for all SFRs in the PP-Module. For any SFRs the ST claims, the corresponding audit events must also be claimed in the table. If the ST does not claim a particular SFR, then there is likewise no expectation that the auditable events for that SFR are required.

Per FAU_STG_EXT.1 in the Base-PP, the TOE must support transfer of the audit data to an external IT entity using a trusted channel.

FAU_GEN.1.2/LiFi

The TSE shall record within the audit data at least the following information:

- a. Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
- b. For each audit event type, based on the auditable event definitions of the functional components included in the PP, PP-Module, functional package, or ST [additional information defined in auditable events tables for each auditable event, where applicable].

Evaluation Activities ▼

FAU_GEN.1/LiFi

TSS

There are no additional TSS evaluation activities for this component.

Guidance

There are no additional Guidance evaluation activities for this component.

Tests

The evaluator shall complete the evaluation activity for FAU_GEN.1 as described in the NDcPP for the auditable events defined in the PP-Module, in addition to the applicable auditable events that are defined in the NDcPP. The evaluator shall also ensure that the administrative actions defined for this PP-Module are appropriately audited.

5.2.3 Cryptographic Support (FCS)

FCS_COP.1/LiFiDataEncryption Cryptographic Operation (Data Encryption and Decryption for LiFi)

FCS_COP.1.1/LiFiDataEncryption

The TSE shall perform [symmetric-key encryption and decryption] in accordance with a specified cryptographic algorithm [**selection:** *Cryptographic Algorithm*] and cryptographic key sizes [**selection:** *Cryptographic Key Sizes*] that meet the following: [**selection:** *List of Standards*]

The following table provides the allowable choices for completion of the selection operations in [FCS_COP.1/LiFiDataEncryption](#). At minimum, AES-GCMP must be selected.

Table 3: Allowable choices for [FCS_COP.1/LiFiDataEncryption](#)

Identifier	Cryptographic Algorithm	Cryptographic Key Sizes	List of Standards
AES-GCMP	AES in Galois Counter Mode Protocol	256 bits	ISO/IEC 18033-3:2010 (Subclause 5.2), NIST SP 800-38D, IEEE 802.11ax-2021
AES-CBC	AES in CBC mode with non-repeating and unpredictable IVs	256 bits	ISO/IEC 18033-3:2010 (Subclause 5.2), ISO/IEC 10116:2017 (Clause 7)
AES-CCMP	AES in CCM mode Protocol	256 bits	ISO/IEC 18033-3:2010 (Subclause 5.2), NIST SP 800-38C, IEEE 802.11-2020
AES-GCM	AES in GCM mode with non-repeating IVs using [selection: <i>deterministic, RBG-based</i>], IV construction; the tag must be of length [selection: 96, 104, 112, 120, 128]	256 bits	ISO/IEC 18033-3:2010 (Subclause 5.2), ISO/IEC 19772:2020 (Clause 10)

Application Note: This requirement includes cryptographic algorithms defined in the Base-PP because it applies to the cryptographic algorithms used specifically in support of LiFi communications, some of which can be used for other purposes. If any of the algorithms claimed in support of LiFi communications are used by the TSF solely for that purpose, it is not necessary to duplicate the claim in the Base-PP. For example, if the TOE implements AES-CBC solely for LiFi communications, the Base-PP requirement FCS_COP.1/SKC does not need to replicate the AES-CBC claim made here. 256-bit GCMP is required in order to comply with FCS_CKM.1/WPA. AES-CCMP, if claimed, is required to use 256-bit keys for IEEE 802.11ax-2021 connections.

Evaluation Activities ▼

FCS_COP.1/LiFiDataEncryption **TSS**

The evaluator shall examine the TSS to verify that it describes how the TOE generates a key based on output from a random bit generator as specified in FCS_RBG.1. The evaluator shall review the TSS to verify that it describes how the functionality described by FCS_RBG.1 is invoked.

The evaluator shall examine the ~~TSS~~ to verify that it identifies the usage and key lifecycle for keys generated using each selected algorithm.

Guidance

The evaluator shall verify that the guidance instructs the administrator how to configure the ~~IQE~~ to generate keys for the selected key generation algorithms for all key types and uses identified in the ~~TSS~~.

Tests

If the ~~IQE~~ claims support for AES-CBC, AES-GCM, or AES-GCMP, the evaluator shall perform the testing defined in the ~~Base-PP~~'s FCS_COP.1/SKC ~~SER~~ for the applicable modes.

AES-CCM Tests

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

128 bit and 256 bit keys

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

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

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

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

Due to the restrictions that IEEE 802.11 specifies for this mode (nonce length of 13 and tag length of 8), it is acceptable to test a subset of the supported lengths as long as the selections fall into the ranges specified above. In this case, the evaluator shall ensure that these are the only supported lengths. To test the generation-encryption functionality of ~~AES~~-CCM, the evaluator shall perform the following four tests:

- Test FCS_COP.1/LiFiDataEncryption:1: For each supported key and associated data length and any supported payload, nonce, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.
- Test FCS_COP.1/LiFiDataEncryption:2: For each supported key and payload length and any supported associated data, nonce, and tag length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values, and obtain the resulting ciphertext.
- Test FCS_COP.1/LiFiDataEncryption:3: For each supported key and nonce length and any supported associated data, payload, and tag length, the evaluator shall supply one key value and 10 associated data, payload, and nonce value 3-tuples, and obtain the resulting ciphertext.
- Test FCS_COP.1/LiFiDataEncryption:4: For each supported key and tag length and any supported associated data, payload, and nonce length, the evaluator shall supply one key value, one nonce value, and 10 pairs of associated data and payload values and, obtain the resulting ciphertext.

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

To test the decryption-verification functionality of AES-CCM, for each combination of supported associated data length, payload length, nonce length, and tag length, the evaluator shall supply a key value and 15 nonce, associated data, and ciphertext 3-tuples, and obtain either a fail result or a pass result with the decrypted payload.

The evaluator shall supply 10 tuples that should fail and 5 that should pass per set of 15.

Additionally, the evaluator shall use tests from the IEEE 802.11-02/362r6 document “Proposed Test vectors for IEEE 802.11 TGi,” dated September 10, 2002, Section 2.1 AES-CCMP Encapsulation Example and Section 2.2 Additional AES-CCMP Test Vectors to verify further the IEEE 802.11-2020 implementation of AES-CCMP.

5.2.4 Identification and Authentication (FIA)

FIA_8021X_EXT.1 802.1X Authentication

FIA_8021X_EXT.1.1

The TSF shall conform to IEEE Standard 802.1X for a Port Access Entity (PAE) in the “Authenticator” role.

FIA_8021X_EXT.1.2

The TSF shall support communications to a RADIUS authentication server conforming to RFCs 2865, 2866, and 3579.

FIA_8021X_EXT.1.3

The TSF shall ensure that no access to its 802.1X controlled port is given to the wireless client prior to successful completion of this authentication exchange.

Application Note: This requirement covers the TOE's role as the authenticator in an 802.1X authentication exchange. If the exchange is completed successfully, the TOE will obtain the PMK from the RADIUS server and perform the four-way handshake with the wireless client (supplicant) to begin 802.11 communications.

As indicated previously, there are at least three communication paths present during the exchange; two with the TOE as an endpoint and one with the TOE acting as a transfer point only. The TOE establishes an Extensible Authentication Protocol (EAP) over Local Area Network (EAPOL) connection with the wireless client as specified in 802.1X-2007. The TOE also establishes (or has established) a RADIUS protocol connection protected either by IPsec or RadSec (TLS) with the RADIUS server. The wireless client and RADIUS server establish an EAP-TLS session (RFC 5216); in this transaction the TOE merely takes the EAP-TLS packets from its EAPOL/RADIUS endpoint and transfers them to the other endpoint. Because the specific authentication method (TLS in this case) is opaque to the TOE, there are no requirements with respect to RFC 5126 in this PP-Module. However, the base RADIUS protocol (2865) has an update (3579) that will need to be addressed in the implementation and evaluation activities. RADIUS accounting is implemented via conformance to RFC 2866.

Additionally, RFC 5080 contains implementation issues that will need to be addressed by developers but which levy no new requirements. The point of performing 802.1X authentication is to provide access to the network (assuming the

authentication was successful and that all 802.11 negotiations are performed successfully); in the terminology of 802.1X, this means the wireless client has access to the "controlled port" maintained by the TOE.

Evaluation Activities ▼

[FIA_8021X_EXT.1](#)

TSS

In order to show that the TSE implements the 802.1X-2010 standard correctly, the evaluator shall ensure that the TSS contains the following information:

- *The sections (clauses) of the standard that the TOE implements*
- *For each identified section, any options selected in the implementation allowed by the standards are specified*
- *For each identified section, any non-conformance is identified and described, including a justification for the non-conformance*

Because the connection to the RADIUS server will be contained in an IPsec or RadSec (TLS) tunnel, the security mechanisms detailed in the RFCs identified in the requirement are not relied on to provide protection for these communications. Consequently, no extensive analysis of the RFCs is required. However, the evaluator shall ensure that the TSS describes the measures (documentation, testing) that are taken by the product developer to ensure that the TOE conforms to the RFCs listed in this requirement.

Guidance

There are no additional Guidance evaluation activities for this component.

Tests

The evaluator shall perform the following tests:

- *Test FIA_8021X_EXT.1:1: The evaluator shall demonstrate that a wireless client has no access to the test network. After successfully authenticating with a RADIUS server through the TOE, the evaluator shall demonstrate that the wireless client does have access to the test network.*
- *Test FIA_8021X_EXT.1:2: The evaluator shall demonstrate that a wireless client has no access to the test network. The evaluator shall attempt to authenticate using an invalid client certificate, such that the EAP-TLS negotiation fails. This should result in the wireless client still being unable to access the test network.*
- *Test FIA_8021X_EXT.1:3: The evaluator shall demonstrate that a wireless client has no access to the test network. The evaluator shall attempt to authenticate using an invalid RADIUS certificate, such that the EAP-TLS negotiation fails. This should result in the wireless client still being unable to access the test network.*

Note: Tests 2 and 3 above are not tests that "EAP-TLS works," although that is a by-product of the test. The test is actually that a failed authentication (under two failure modes) results in denial of access to the network, which demonstrates the enforcement of [FIA_8021X_EXT.1.3](#).

FIA_UAU.6 Re-authenticating

FIA_UAU.6.1

The TSE shall re-authenticate the user under the conditions [when the user changes their password, [selection: following TSE-initiated session locking, [assignment: other conditions], no other conditions]].

Evaluation Activities ▼

[FIA_UAU.6](#)

TSS

There are no additional TSS evaluation activities for this component.

Guidance

There are no additional Guidance evaluation activities for this component.

Tests

The evaluator shall attempt to change their password as directed by the operational guidance. While making this attempt, the evaluator shall verify that re-authentication is required.

If other re-authentication conditions are specified, the evaluator shall cause those conditions to occur and verify that the TSF re-authenticates the authenticated user.

5.2.5 Security Management (FMT)

FMT_SMF.1/LiFi Specification of Management Functions

FMT_SMF.1.1/LiFi

The TSF shall be capable of performing the following management functions:
[configuration of the security policy for each wireless network, including:

- Security type
- Client credentials to be used for authentication
- [**selection**: authentication protocol, Service Set Identifier (SSID) if the SSID is broadcast, no other security policy elements]

].

Evaluation Activities ▼

[FMT_SMF.1/LiFi](#)

TSS

The evaluator shall examine the TSS to verify that it describes the security policy configuration in terms of the attributes claimed in the SER.

The evaluator shall also confirm that the TSS includes a description of how connection attempts from clients that are not operating on an approved security type are handled.

Guidance

The evaluator shall examine the operational guidance to verify that it includes procedures for configuring the security policy of the wireless network including all attributes claimed in the ST.

Tests

The evaluator shall perform the following tests:

- Test FMT_SMF.1/LiFi:1: For each security type specified in the TSS, configure the network to the approved security type and verify that the client can establish a connection.

Maintaining the same SSID, change the security type of the client to a non-approved security type and attempt to establish a connection. Verify that the connection was unsuccessful.

- Test FMT_SMF.1/LiFi:2: For each authentication protocol specified in the TSS, configure the network accordingly per the AGD. Verify that the client connection attempt is successful when using the correct client credentials and that the connection is unsuccessful when incorrect authentication credentials are used.
- Test FMT_SMF.1/LiFi:3: [Conditional: SSID is broadcast] The evaluator shall configure the SSID to be broadcast. Using a network sniffing tool, the evaluator shall capture a beacon frame and confirm that the SSID is included. The evaluator shall then configure the SSID to be hidden. Using a network sniffing tool, the evaluator shall capture a beacon frame and confirm that the SSID is not listed.

FMT_SMR_EXT.1 No Administration from Client

FMT_SMR_EXT.1.1

The TSF shall ensure that the ability to remotely administer the TOE from a wireless client shall be disabled by default.

Evaluation Activities ▼

FMT_SMR_EXT.1

TSS

There are no additional TSS evaluation activities for this component.

Guidance

The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration. The evaluator shall confirm that the TOE does not permit remote administration from a wireless client by default.

Tests

The evaluator shall demonstrate that after configuring the TOE for first use from the operational guidance, it is possible to establish an administrative session with the TOE on the “wired” portion of the device. The evaluator shall then demonstrate that an identically configured wireless client that can successfully connect to the TOE cannot be used to perform administration.

5.2.6 Protection of the TSF (FPT)

FPT_ACF_EXT.1 Access Controls

FPT_ACF_EXT.1.1

The TSF shall implement access controls which prohibit unprivileged users from modifying:

- Kernel and its drivers and modules
- Security audit logs
- Shared libraries
- System executables
- System configuration files
- [assignment: other objects].

FPT_ACF_EXT.1.2

The TSS shall implement access controls which prohibit unprivileged users from reading:

- Security audit logs
- System-wide credential repositories
- [assignment: list of other objects].

Application Note: "Credential repositories" refer, in this case, to structures containing cryptographic keys or passwords.

Evaluation Activities ▼

[FPT_ACF_EXT.1.1](#)

TSS

The evaluator shall confirm that the TSS specifies the locations of kernel drivers and modules, security audit logs, shared libraries, system executables, and system configuration files. Every file does not need to be individually identified, but the system's conventions for storing and protecting such files must be specified.

Guidance

There are no additional Guidance evaluation activities for this element.

Tests

For these tests the evaluator shall use appropriate development environment (e.g., a Virtual Machine) and development tools (debuggers, simulators, etc.), including enabling root shell access as needed. The evaluator shall create an unprivileged user account. Using this account, the evaluator shall ensure that the following tests result in a negative outcome (i.e., the action results in the OS denying the evaluator permission to complete the action):

- Test FPT_ACF_EXT.1.1:1: The evaluator shall attempt to modify all kernel drivers and modules.
- Test FPT_ACF_EXT.1.1:2: The evaluator shall attempt to modify all security audit logs generated by the logging subsystem.
- Test FPT_ACF_EXT.1.1:3: The evaluator shall attempt to modify all shared libraries that are used throughout the system.
- Test FPT_ACF_EXT.1.1:4: The evaluator shall attempt to modify all system executables.
- Test FPT_ACF_EXT.1.1:5: The evaluator shall attempt to modify all system configuration files.
- Test FPT_ACF_EXT.1.1:6: The evaluator shall attempt to modify any additional components selected.

[FPT_ACF_EXT.1.2](#)

TSS

There are no additional TSS evaluation activities for this element.

Guidance

There are no additional Guidance evaluation activities for this element.

Tests

For these tests the evaluator shall use appropriate development environment (e.g., a Virtual Machine) and development tools (debuggers, simulators, etc.), including enabling root shell access as needed. The evaluator shall create an unprivileged user account. Using this account, the evaluator shall ensure that the following tests result in a negative outcome (i.e., the action results in

the OS denying the evaluator permission to complete the action):

- Test FPT_ACF_EXT.1.2:1: The evaluator shall attempt to read security audit logs generated by the auditing subsystem.
- Test FPT_ACF_EXT.1.2:2: The evaluator shall attempt to read system-wide credential repositories.
- Test FPT_ACF_EXT.1.2:3: The evaluator shall attempt to read any other object specified in the assignment.

FPT_ASLR_EXT.1 Address Space Layout Randomization

FPT_ASLR_EXT.1.1

The TSSF shall always randomize process address space memory locations with [selection: 8, [assignment: number greater than 8]] bits of entropy except for [assignment: list of explicit exceptions].

Evaluation Activities ▼

[FPT_ASLR_EXT.1](#)

TSS

The evaluator shall examine the TSS to verify that it identifies the number of entropy bits used for ASLR and what exceptions there are to this, if any.

Guidance

There are no additional Guidance evaluation activities for this component.

Tests

For these tests the evaluator shall use appropriate development environment (e.g., a Virtual Machine) and development tools (debuggers, simulators, etc.), including enabling root shell access as needed. The evaluator shall select three executables included with the TSSF. If the TSSF includes a web server it must be selected. The evaluator shall prioritize executables that are UDP or TCP listening network services. For each of these executables, the evaluator shall launch the same executable twice and inspect the memory map using an appropriate tool or interface (e.g. /proc/\$PID/maps on a Linux-based OS). The evaluator shall ensure that no memory mappings are placed in the same location. If the rare chance occurs that two mappings are the same for a single executable and not the same for another chosen executable, the evaluator shall repeat the test with that executable to verify that in the second test the mappings are different. This test should also be completed on the same hardware. The evaluator shall also inspect the executables to verify the fPIE compile flag is enabled. The evaluator shall read /proc/kallsyms and note the _stext address. The evaluator shall then completely reboot the TSSF and re-read /proc/kallsyms and in verify that the addresses differ. In the rare event KASLR is on and the addresses are the same the evaluator shall reboot and verify again.

FPT_SBOP_EXT.1 Stack Buffer Overflow Protection

FPT_SBOP_EXT.1.1

The TSSF shall [selection: employ stack-based buffer overflow protections, not store parameters or variables in the same data structures as control flow values].

Application Note: Many OSES store control flow values (i.e., return addresses) in stack data structures that also contain parameters and variables. For these OSES, it is expected that most of the OS, to include the kernel, libraries, and application software from the OS vendor be compiled with stack-based buffer overflow protection enabled. OSES that store parameters and variables separately from control flow values do not need additional stack protections.

Evaluation Activities ▼

FPT_SBOP_EXT.1

TSS

The evaluator shall examine the TSS to verify that it describes how stack-based buffer overflow protection is implemented, or how the TOE is designed in such a way that these protections are not explicitly needed.

Guidance

There are no additional Guidance evaluation activities for this component.

Tests

For stack-based OSES, the evaluator shall determine that the TSS contains a description of stack-based buffer overflow protections used by the OS. These are referred to by a variety of terms. These include, but are not limited to, tagging, stack cookie, stack guard, and stack canaries. The TSS must include a rationale for any binaries that are not protected in this manner. The evaluator shall also preform the following test. For this test the evaluator shall use an appropriate development environment (e.g., a Virtual Machine) and development tools (debuggers, simulators, etc.), including enabling root shell access as needed.

- *Test FPT_SBOP_EXT.1:1: The evaluator shall inventory the kernel, libraries, and application binaries to determine those that do not implement stack-based buffer overflow protections. This list should match up with the list provided in the TSS.*

For OSES that store parameters or variables separately from control flow values, the evaluator shall verify that the TSS describes what data structures control values, parameters, and variables are stored. The evaluator shall also ensure that the TSS includes a description of the safeguards that ensure parameters and variables do not intermix with control flow values.

FPT_W^X_EXT.1 Write XOR Execute Memory Pages

FPT_W^X_EXT.1.1

The TSE shall prevent allocation of any memory region with both write and execute permissions except for [assignment: list of exceptions].

Application Note: Requesting a memory mapping with both write and execute permissions subverts the platform protection provided by DEP. If the OS provides no exceptions (such as for just-in-time compilation), then "no exceptions" should be indicated in the assignment. Full realization of this requirement requires hardware support, but this is commonly available.

Evaluation Activities ▼

[FPT_W^X_EXT.1](#)

TSS

The evaluator shall inspect the vendor-provided developer documentation and verify that no memory-mapping can be made with write and execute permissions except for the cases listed in the assignment.

Guidance

There are no additional Guidance evaluation activities for this component.

Tests

- *Test FPT_W^X_EXT.1:1: The evaluator shall acquire or construct a test program which attempts to allocate memory that is both writable and executable. The evaluator shall run the program and confirm that it fails to allocate memory that is both writable and executable.*
- *Test FPT_W^X_EXT.1:2: The evaluator shall acquire or construct a test program which allocates memory that is executable and then subsequently requests additional write or modify permissions on that memory. The evaluator shall run the program and confirm that at no time during the lifetime of the process is the memory both writable and executable.*
- *Test FPT_W^X_EXT.1:3: The evaluator shall acquire or construct a test program which allocates memory that is writable and then subsequently requests additional execute permissions on that memory. The evaluator shall run the program and confirm that at no time during the lifetime of the process is the memory both writable and executable.*

5.2.7 TOE Access (FTA)

FTA_TSE.1 TOE Session Establishment

FTA_TSE.1.1

The TSS shall be able to deny session establishment **of a wireless client session** based on [TOE interface, time, day, [*selection*: [*assignment*: other attributes], no other attributes]].

Application Note: The “TOE interface” can be specified in terms of the device in the TOE that the LiFi client is connecting to (e.g., specific LiFi APs). “time” and “day” refer to time-of-day and day-of-week, respectively.

The assignment is to be used by the ST author to specify additional attributes on which denial of session establishment can be based.

Evaluation Activities ▼

[FTA_TSE.1](#)

TSS

The evaluator shall examine the TSS to determine that all of the attributes on which a client session can be denied are specifically defined.

Guidance

The evaluator shall examine the operational guidance to determine that it contains guidance for configuring each of the attributes identified in the TSS.

Tests

For each supported attribute, the evaluator shall perform the following test:

- *Test FTA_TSE.1:1: The evaluator shall successfully establish a client session with a wireless client. The evaluator shall then follow the operational guidance to configure the system so that the client's access is denied based on a specific value of the attribute. The evaluator shall then attempt to establish a session in contravention to the attribute setting (for instance, the client is denied network access based on the TOE interface that it is connecting to, or that the client is denied access based on the time-of-day or day-of-week it is attempting connection on). The evaluator shall observe that the access attempt fails.*

5.2.8 Trusted Path/Channels (FTP)

FTP_ITC.1/8021X Inter-TSF Trusted Channel (802.1X)

FTP_ITC.1.1/8021X

The TSF shall provide a communication channel **using IEEE 802.1X and [selection: IPsec, RADIUS over TLS]** between itself and **an 802.1X authentication server** that is logically distinct from other communication channels and provides assured identification of its endpoints and protection of the channel data from modification or disclosure.

Application Note: This requirement has been iterated from its definition in the NDcPP to mandate the communications protocols and environmental components that a LiFi AS must use for 802.1X. IPsec or RADIUS over TLS (commonly known as "RadSec") is required at least for communications with the 802.1X authentication server. The requirement implies that not only are communications protected when they are initially established, but also on resumption after an outage.

The IT entity of "802.1X authentication server" is distinct from "authentication server" as specified in FTP_ITC.1 in the NDcPP because the latter may be used for administrator authentication rather than authorization of LiFi clients.

If IPsec is selected in FTP_ITC.1.1, then FCS_IPSEC_EXT.1 from the NDcPP must be claimed. If RADIUS over TLS is selected in FTP_ITC.1.1, then FCS_RADSEC_EXT.1 in this PP-Module must be claimed, as well as FCS_TLSC_EXT.1 from the Functional Package for TLS.

FTP_ITC.1.2/8021X

The TSF shall permit [*the TSF*] to initiate communication via the trusted channel.

FTP_ITC.1.3/8021X

The TSF shall initiate communication via the trusted channel for [*802.1X authentication*].

Evaluation Activities ▼

FTP_ITC.1/8021X

The evaluator shall perform the following activities in addition to those required by the NDcPP:

ISS

The evaluator shall examine the ISS to determine that for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the

allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST.

Guidance

The evaluator shall confirm that the guidance documentation contains instructions for establishing the allowed protocols with each authorized IT entity and that it contains recovery instructions should a connection be unintentionally broken.

Tests

The evaluator shall perform the following tests:

- Test FTP_ITC.1/8021X:1: The evaluator shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the guidance documentation and ensuring that communication is successful.
- Test FTP_ITC.1/8021X:2: For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the guidance documentation to ensure that the communication channel can be initiated from the TOE.
- Test FTP_ITC.1/8021X:3: The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data is not sent in plaintext.
- Test FTP_ITC.1/8021X:4: The evaluator shall, for each protocol associated with each authorized IT entity tested during test 1, physically interrupt an established connection. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected.

5.3 TOE Security Functional Requirements Rationale

The following rationale provides justification for each SFR for the TOE, showing that the SFRs are suitable to address the specified threats:

Table 4: SFR Rationale

Threat	Addressed by	Rationale
T.DATA_ INTEGRITY	FCS_COP.1/LiFiDataEncryption	Mitigates the threat by defining implementation of secure cryptographic integrity algorithms.
	FCS_CKM.1/WPA (implementation-dependent)	Mitigates the threat by generating a secure key used to encrypt and protect wireless traffic using WPA.
	FCS_CKM.2/DISTRIB (implementation-dependent)	Mitigates the threat by securely distributing 802.11 keys to clients so that they are not disclosed, maintaining the integrity of the data the key is used to protect.
	FCS_CKM.2/GTK (implementation-dependent)	Mitigates the threat by securely distributing group temporal keys to clients so that they are not disclosed, maintaining the integrity of the data the key is used to protect.

	FCS_CKM.2/PMK (implementation-dependent) FTP_ITC.1/Client (implementation-dependent) FTP_ITC.1/GvIc (implementation-dependent)	Mitigates the threat by securely receiving a key from an authentication server so that it is not disclosed, maintaining the integrity of the data the key is used to protect. Mitigates the threat by providing a secure channel for LiFi client traffic that protects the confidentiality and integrity of the data. Mitigates the threat by providing a secure channel for LiFi client traffic that protects the confidentiality and integrity of the data.
T.NETWORK_DISCLOSURE	FCS_COP.1/LiFiDataEncryption FIA_UAU.6 FTA_TSE.1 FTP_ITC.1/8021X FCS_CKM.1/WPA (implementation-dependent) FCS_CKM.2/DISTRIB (implementation-dependent) FCS_CKM.2/GTK (implementation-dependent) FCS_CKM.2/PMK (implementation-dependent) FTP_ITC.1/Client (implementation-dependent) FTP_ITC.1/GvIc (implementation-dependent) FCS_RADSEC_EXT.1 (selection-based) FCS_RADSEC_EXT.2 (selection-based)	Mitigates the threat by defining implementation of secure cryptographic confidentiality algorithms. Mitigates the threat by re-authenticating users when specified criteria are met. Mitigates the threat by denying sessions based on undesirable characteristics. Mitigates the threat by enforcing the use of authentication to access the network. Mitigates the threat by using a secure key generation function for WPA functionality. Mitigates the threat by securely distributing 802.11 keys to clients. Mitigates the threat by securely distributing GTK keys to clients. Mitigates the threat by securely receiving a key from an authentication server. Mitigates the threat by providing a secure channel for LiFi client traffic. Mitigates the threat by providing a secure channel for LiFi client traffic. Mitigates the threat by using a secure RADIUS implementation to authenticate clients. Mitigates the threat by using a secure TLS configuration for RADIUS over TLS.
T.NETWORK_ACCESS	FAU_GEN.1/LiFi FIA_8021X_EXT.1	Mitigates the threat by generating audit data that will detect access to the network. Mitigates the threat by enforcing authentication on access attempts.

	FIA_UAU.6	Mitigates the threat by re-authenticating users when specified criteria are met.
	FMT_SMF.1/LiFi	Mitigates the threat by defining management functions that must be protected.
	FMT_SMR_EXT.1	Mitigates the threat by preventing remote administration functions from being exercised from a wireless client.
	FTA_TSE.1	Mitigates the threat by denying sessions based on undesirable characteristics.
	FTP_ITC.1/8021X	Mitigates the threat by authenticating clients before allowing access to the controlled network.
	FIA_PSK_EXT.1 (implementation-dependent)	Mitigates the threat by defining the use of a PSK to authenticate clients.
	FCS_RADSEC_EXT.1 (selection-based)	Mitigates the threat by using a secure RADIUS implementation to authenticate clients.
	FCS_RADSEC_EXT.2 (selection-based)	Mitigates the threat by using a secure TLS configuration for RADIUS over TLS.
T.NETWORK_ATTACK	FAU_GEN.1/LiFi	Mitigates the threat by generating audit data that will detect TSF failures.
	FPT_ACF_EXT.1	Mitigates the threat by limiting access that could be obtained on the system as a result of successful network attack.
	FPT_ASLR_EXT.1	Mitigates the threat by preventing network attacks that rely on predictable program memory layout.
	FPT_SBOP_EXT.1	Mitigates the threat by preventing network attacks that rely on overwriting stack memory to execute arbitrary code.
	FPT_W^X_EXT.1	Mitigates the threat by preventing network attacks that rely on writing arbitrary executable code to memory and subsequently executing it.
T.REPLAY_ATTACK	FCS_COP.1/LiFiDataEncryption	Mitigates the threat by defining implementation of secure cryptographic integrity algorithms that prevent the disclosure of data that would be potentially suitable for replay attempts.
	FTP_ITC.1/8021X	Mitigates the threat by authenticating clients before allowing access to the controlled network and preventing replay of previous authentication sessions.
	FIA_UAU.6	Mitigates the threat by re-authenticating users when specified criteria are met, which can include

	if a specified time has passed or other indicators of stale authentication data.
FTA_TSE.1	Mitigates the threat by denying sessions based on undesirable characteristics, which can include a detection of replay or other indicators of stale authentication data.
FCS_CKM.1/WPA (implementation-dependent)	Mitigates the threat by using a secure key generation function for WPA functionality that is not subject to replay or guessing.
FCS_CKM.2/DISTRIB (implementation-dependent)	Mitigates the threat by securely distributing 802.11 keys to clients so that they are not disclosed, maintaining confidentiality and preventing replay.
FCS_CKM.2/GTK (implementation-dependent)	Mitigates the threat by securely distributing GTK keys to clients so that they are not disclosed, maintaining confidentiality and preventing replay.
FCS_CKM.2/PMK (implementation-dependent)	Mitigates the threat by securely receiving a key from an authentication server so that it is not disclosed, maintaining confidentiality and preventing replay.
FTP_ITC.1/Client (implementation-dependent)	Mitigates the threat by providing a secure channel for LiFi client traffic that prevents the use of replayed data.
FTP_ITC.1/Gvlc (implementation-dependent)	Mitigates the threat by providing a secure channel for LiFi client traffic that prevents the use of replayed data.
FCS_RADSEC_EXT.1 (selection-based)	Mitigates the threat by using a secure RADIUS implementation to authenticate clients, which protects authentication data from replay attacks.
FCS_RADSEC_EXT.2 (selection-based)	Mitigates the threat by using a secure TLS configuration for RADIUS over TLS , which provides protection against replay and data modification.

6 Consistency Rationale

6.1 Collaborative Protection Profile for Network Device

6.1.1 Consistency of TOE Type

When this PP-Module extends the NDcPP, the TOE type for the overall TOE is still a network device. This PP-Module just defines the TOE as a specific type of network device with functional capabilities distinct to that type.

6.1.2 Consistency of Security Problem Definition

Table 5: Consistency of Security Problem Definition (NDcPP base)

PP-Module Threat, Assumption, OSP	Consistency Rationale
T.DATA_INTEGRITY	This threat is a specific type of failure that may result from successful exploitation of the T.WEAK_CRYPTOGRAPHY threat defined by the Base-PP. It is an extension of the Base-PP threat for communications that are specific to this PP-Module.
T.NETWORK_DISCLOSURE	This threat extends the security problem defined by the Base-PP to include the threat of a malicious entity in an untrusted network interacting with a protected entity in a trusted network. This is not addressed in the Base-PP because not all network devices are responsible for facilitating communications between separate networks. This threat is also consistent with the T.UNTRUSTED_COMMUNICATION_CHANNELS threat defined by the Base-PP because compromise of data in transit is one potential way this threat may be exploited.
T.NETWORK_ACCESS	This threat extends the security problem defined by the Base-PP to include the threat of a malicious entity in an untrusted network interacting with a protected entity in a trusted network. This is not addressed in the Base-PP because not all network devices are responsible for facilitating communications between separate networks.
T.NETWORK_ATTACK	This threat extends the security problem defined by the Base-PP to define a specific manifestation of T.SECURITY_FUNCTIONALITY_FAILURE.
T.REPLAY_ATTACK	This threat is a specific type of failure that may result from successful exploitation of the T.UNAUTHORIZED_ADMINISTRATOR_ACCESS and T.UNTRUSTED_COMMUNICATIONS_CHANNELS threats defined by the Base-PP. It is an extension of the Base-PP threat for communications that are specific to this PP-Module.
A.NON_SECURITY_FUNCTIONS	This is consistent with the Base-PP because the notion of exact PP conformance accepts the case where non-TSF functionality exists within

the physical boundary of the ~~TOE~~ if there are no ~~PP~~ requirements to evaluate it.

A.MANAGED_CONFIG

This is consistent with A.TRUSTED_ADMINISTRATOR in the ~~Base-PP~~ by defining specific mechanisms by which the trusted administrator is defined.

A.THIRD_PARTY_SOFTWARE

This is consistent with the ~~Base-PP~~ because the existence of a trusted update mechanism allows for the possibility that the ~~TOE~~ may have implementation flaws in need of fixing.

A.NETWORK_CONNECTIVITY

This is consistent with the ~~Base-PP~~ because a network device inherently requires network connectivity and the ~~Base-PP~~ similarly does not treat external entities as automatically trusted (hence the use of trusted channels to authenticate them).

A.PROPER_USER

This is consistent with the expectations of A.TRUSTED_ADMINISTRATOR in the ~~Base-PP~~ by defining administrators as non-hostile but allowing for the possibility of unintentional error.

6.1.3 Consistency of OE Objectives

Table 6: Consistency of ~~OE~~ Objectives (NDcPP base)

PP-Module OE Objective	Consistency Rationale
OE .NON_SECURITY_FUNCTIONS	This is consistent with the Base-PP because the notion of exact PP conformance accepts the case where non-TSF functionality exists within the physical boundary of the TOE if there are no PP requirements to evaluate it.
OE .MANAGED_CONFIG	This is consistent with A.TRUSTED_ADMINISTRATOR in the Base-PP by defining specific mechanisms by which the trusted administrator is defined.
OE .THIRD_PARTY_SOFTWARE	This is consistent with the Base-PP because the existence of a trusted update mechanism allows for the possibility that the TOE may have implementation flaws in need of fixing.
OE .NETWORK_CONNECTIVITY	This is consistent with the Base-PP because a network device inherently requires network connectivity and the Base-PP similarly does not treat external entities as automatically trusted (hence the use of trusted channels to authenticate them).
OE .PROPER_USER	This is consistent with the expectations of A.TRUSTED_ADMINISTRATOR in the Base-PP by defining administrators as non-hostile but allowing for the possibility of unintentional error.

6.1.4 Consistency of Requirements

This ~~PP-Module~~ identifies several ~~SFRs~~ from the NDcPP that are needed to support LiFi Access System functionality. This is considered to be consistent because the functionality provided by the NDcPP is being used for

its intended purpose. The rationale for why this does not conflict with the claims defined by the NDcPP are as follows:

Table 7: Consistency of Requirements (NDcPP base)

<u>PP-Module</u> Requirement	Consistency Rationale
Modified <u>SERs</u>	
This <u>PP-Module</u> does not modify any requirements when the <u>NDcPP</u> is the base.	
Additional <u>SERs</u>	
This <u>PP-Module</u> does not add any requirements when the <u>NDcPP</u> is the base.	
Mandatory <u>SERs</u>	
<u>FAU_GEN.1/LiFi</u>	This <u>SER</u> iterates the <u>FAU_GEN.1</u> <u>SER</u> defined in the <u>Base-PP</u> to define auditable events for the functionality that is specific to this <u>PP-Module</u> .
<u>FCS_COP.1/LiFiDataEncryption</u>	This <u>SER</u> defines cryptographic behavior for functions outside the scope of what the <u>Base-PP</u> originally defined.
<u>FIA_8021X_EXT.1</u>	This <u>SER</u> defines support for 802.1X communications, which is a logical interface that extends the scope of what the <u>Base-PP</u> originally defined.
<u>FIA_UAU.6</u>	This <u>SER</u> defines support for re-authentication of wireless users, which are a type of subject beyond the scope of what the <u>Base-PP</u> originally defined.
<u>FMT_SMF.1/LiFi</u>	This <u>SER</u> defines additional management functionality that is specific to the Module's product type and would therefore not be expected to be present in the <u>Base-PP</u> .
<u>FMT_SMR_EXT.1</u>	This <u>SER</u> specifies that remote administration occurs over a dedicated channel, which does not conflict with the <u>Base-PP</u> .
<u>FPT_ACF_EXT.1</u>	This <u>SER</u> specifies that unprivileged users are prohibited from reading or modifying certain parts of the system that could cause it to operate in an insecure state, which does not conflict with the <u>Base-PP</u> .
<u>FPT_ASLR_EXT.1</u>	This <u>SER</u> enforces specific low-level protections on the software running on the <u>TOE</u> that does not conflict with any other security functionality. The ability of the <u>TOE</u> to implement <u>ASLR</u> measures does not prevent its code from performing any required function.
<u>FPT_SBOP_EXT.1</u>	This <u>SER</u> enforces specific low-level protections on the software running on the <u>TOE</u> that does not conflict with any other security functionality. The ability of the <u>TOE</u> to implement stack-based overflow protection does not prevent its code from performing any required function.
<u>FPT_W^X_EXT.1</u>	This <u>SER</u> enforces specific low-level protections on the software running on the <u>TOE</u> that does not conflict with any other security functionality. The ability of the <u>TOE</u> to prevent executable memory from also being writable does not prevent its code from performing any required function.

FTA_TSE.1 This SFR applies restrictions on establishment of wireless communications, which is a logical interface that extends the scope of what the Base-PP originally defined.

FTP_ITC.1/8021X This SFR iterates the FTP_ITC.1 SFR defined in the Base-PP to define trusted communication channels for the functionality that is specific to this PP-Module.

Optional SFRs

This PP-Module does not define any Optional requirements.

Objective SFRs

This PP-Module does not define any Objective requirements.

Implementation-dependent SFRs

FCS_CKM.1/WPA This SFR defines additional cryptographic functionality not defined in the Base-PP, but it implements this using the DRBG mechanism already defined in the Base-PP.

FCS_CKM.2/GTK This SFR defines additional cryptographic functionality not defined in the Base-PP that is used for functionality outside the original scope of the Base-PP.

FCS_CKM.2/PMK This SFR defines additional cryptographic functionality not defined in the Base-PP that is used for functionality outside the original scope of the Base-PP.

FIA_PSK_EXT.1 This SFR defines parameters for pre-shared key generation. The Base-PP supports pre-shared keys as a potential authentication method for IPsec. This PP-Module does not prevent this from being used but does define restrictions on how pre-shared keys may be generated and what constitutes an acceptable key. This may also be used for RadSec, which is outside the original scope of the Base-PP.

FTP_ITC.1/Client This SFR iterates the FTP_ITC.1 SFR defined in the Base-PP to define trusted communication channels for the functionality that is specific to this PP-Module.

FCS_CKM.2/DISTRIB This SFR defines an additional use for the cryptographic and self-protection mechanisms defined in the Base-PP.

FTP_ITC.1/Gvlc This SFR iterates the FTP_ITC.1 SFR defined in the Base-PP to define trusted communication channels for the functionality that is specific to this PP-Module.

Selection-based SFRs

FCS_RADSEC_EXT.1 This SFR defines the implementation of RadSec and the peer authentication method that it uses. This relies on the TLS requirements defined by the Base-PP and may also use the X.509v3 certificate validation methods specified in the Base-PP, depending on the selected peer authentication method.

FCS_RADSEC_EXT.2

This ~~SFR~~ defines the implementation of RadSec when pre-shared key authentication is used. This functionality is outside the original scope of the ~~Base-PP~~, but it relies on the ~~TLS~~ client protocol implementation, cryptographic algorithms, and random bit generation functions defined by the ~~Base-PP~~.

Appendix A - Optional SFRs

A.1 Strictly Optional Requirements

This PP-Module does not define any Strictly Optional SFRs or SARs.

A.2 Objective Requirements

This PP-Module does not define any Objective SFRs.

A.3 Implementation-dependent Requirements

A.3.1 Auditable Events for Implementation-Dependent SFRs

Table 8: Auditable Events for Implementation-dependent Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FCS_CKM.1/WPA	No events specified	N/A
FCS_CKM.2/DISTRIB	No events specified	N/A
FCS_CKM.2/GTK	No events specified	N/A
FCS_CKM.2/PMK	No events specified	N/A
FIA_PSK_EXT.1	No events specified	N/A
FTP_ITC.1/Client	Failed attempts to establish a trusted channel (including IEEE 802.11)	Identification of the initiator and target of channel
	Detection of modification of channel data	Identification of the initiator and target of channel
FTP_ITC.1/Gvlc	Failed attempts to establish a trusted channel	Identification of the initiator and target of channel
	Detection of modification of channel data	Identification of the initiator and target of channel

A.3.2 SFRs for 802.11bb Implementations

FCS_CKM.1/WPA Cryptographic Key Generation (Symmetric Keys for WPA2 Connections)

This component must be included in the ST if the TQE implements any of the following features:

- *802.11bb Support*

FCS_CKM.1.1/WPA

The TSE shall generate **symmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [**selection:** *Cryptographic Key Generation Algorithm*] and specified cryptographic key sizes [**selection:** *Cryptographic Key Sizes*] that meet the following: [**selection:** *List of standards*]

The following table provides the allowable choices for completion of the selection operations of [FCS_CKM.1/WPA](#). At minimum, PRF-384 must be selected.

Table 9: Allowable Choices for [FCS_CKM.1/WPA](#)

Identifier	Cryptographic Key Generation Algorithm	Cryptographic Key Sizes	List of standards
PRF-384	Pseudorandom Function using a random bit generator as specified in FCS_RBG.1 (from Base-PP)	384 bits	IEEE 802.11-2020
PRF-512	Pseudorandom Function using a random bit generator as specified in FCS_RBG.1 (from Base-PP)	512 bits	IEEE 802.11-2020
PRF-704	Pseudorandom Function using a random bit generator as specified in FCS_RBG.1 (from Base-PP)	704 bits	IEEE 802.11-2020, IEEE 802.11ax-2021

Application Note: The cryptographic key derivation algorithm required by IEEE 802.11-2020 (Section 12.7.1.2) and verified in WPA2 certification is PRF-384, which uses the HMAC-SHA-1 function and outputs a 384-bit key. The use of GCMP is defined in IEEE 802.11ax-2021 (Section 12.5.5) and requires a Key Derivation Function (KDF) based on HMAC-SHA-256 (for 128-bit symmetric keys) or HMAC-SHA-384 (for 256-bit symmetric keys). This KDF outputs 704 bits. This requirement applies only to the keys that are generated or derived for the communications between the AP and the client once the client has been authenticated. It refers to the derivation of the Group Temporal Key (GTK), through the Random Bit Generator (RBG) specified in this ~~PP-Module~~, as well as the derivation of the Pairwise Transient Key (PTK) from the Pairwise Master Key (PMK), which is done using a random value generated by the RBG specified in this ~~PP-Module~~, the HMAC function as specified in this ~~PP-Module~~, as well as other information. This is specified in IEEE 802.11-2020, primarily in chapter 12. FCS_RBG.1 is defined in the ~~NDcPP~~.

Evaluation Activities ▼

The cryptographic primitives will be verified through evaluation activities specified elsewhere in this PP-Module. The evaluator shall verify that the TSS describes how the primitives defined and implemented by this PP-Module are used by the TOE in establishing and maintaining secure connectivity to the wireless clients. This description will include how the GTK and PTK are generated or derived. The TSS shall also provide a description of the developer's methods of assuring that their implementation conforms to the cryptographic standards; this includes not only testing done by the developing organization, but also proof of third-party testing that is performed (e.g., WPA2 certification). The evaluator shall ensure that the description of the testing methodology is of sufficient detail to determine the extent to which the details of the protocol specifics are tested.

Guidance

There are no additional Guidance evaluation activities for this component.

Tests

The evaluator shall perform the following test using a packet sniffing tool to collect frames between the TOE and a wireless client:

Step 1: The evaluator shall identify the channel used by the TOE and configure the sniffer to sniff only on that channel (i.e., lock the sniffer on the selected channel). The sniffer should also be configured to filter on the MAC address of the TOE and client.

Step 2: The evaluator shall configure the TOE to communicate with a LiFi client using IEEE 802.11-2020 and a 256-bit (64 hex values 0-f) pre-shared key, setting up the connections as described in the operational guidance. The pre-shared key is only used for testing.

Step 3: The evaluator shall start the sniffing tool, initiate a connection between the TOE and LiFi client, and allow the TOE to authenticate, associate, and successfully complete the four-way handshake with the client.

Step 4: The evaluator shall set a timer for one minute, at the end of which the evaluator shall disconnect the client from the TOE and stop the sniffer.

Step 5: The evaluator shall identify the four-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK from the four-way handshake frames and pre-shared key as specified in IEEE 802.11-2020.

Step 6: The evaluator shall select the first data frame from the captured packets that was sent between the client and TOE after the four-way handshake successfully completed and without the frame control value 0x4208 (the first two bytes are 08 42). The evaluator shall use the PTK to decrypt the data portion of the packet as specified in IEEE 802.11-2020 and verify that the decrypted data contains ASCII-readable text.

Step 7: The evaluator shall repeat Step 6 for the next two data frames between the TOE and client, and without frame control value 0x4208.

Additionally, the evaluator shall test the PRF function using the test vectors from:

- Section 2.4 "The PRF Function – PRF (key, prefix, data, length)" of the IEEE 802.11-02/362r6 document "Proposed Test vectors for IEEE 802.11 TGi" dated September 10, 2002
- Annex J.3 "PRF reference implementation and test vectors" of IEEE 802.11-2020

FCS_CKM.2/GTK Cryptographic Key Distribution (GTK)

This component must be included in the ST if the TOE implements any of the following features:

- **802.11bb Support**

FCS_CKM.2.1/GTK

The TSE shall distribute **Group Temporal Key (GTK)** in accordance with a specified cryptographic key distribution method: [**selection**: *AES Key Wrap in an EAPOL-Key frame, AES Key Wrap with Padding in an EAPOL-Key frame*] that meets the following: [*NIST SP 800-38F, IEEE 802.11-2020 for the packet format and timing considerations*] **and does not expose the cryptographic keys.**

Application Note: This requirement applies to the GTK that is generated by the TOE for use in broadcast and multicast messages to clients to which it is connected. 802.11-2020 specifies the format for the transfer as well as the fact that it must be wrapped by the AES Key Wrap method specified in NIST SP 800-38F.

Evaluation Activities ▼

FCS_CKM.2/GTK

TSS

The evaluator shall check the TSS to ensure that it describes how the GTK is wrapped prior to being distributed using the AES implementation specified in this PP-Module, and also how the GTKs are distributed when multiple clients connect to the TOE.

Guidance

There are no additional Guidance evaluation activities for this component.

Tests

The evaluator shall perform the following test using a packet sniffing tool to collect frames between a wireless client and the TOE (which may be performed in conjunction with the evaluation activity for FCS_CKM.1/PMK).

To fully test the broadcast and multicast functionality, these steps will be performed as the evaluator connects multiple clients to the TOE. The evaluator shall ensure that GTKs established are sent to the appropriate participating clients.

Step 1: The evaluator shall identify the channel used by the TOE and configure the sniffer to sniff only on that channel (i.e., lock the sniffer on the selected channel). The sniffer should also be configured to filter on the MAC address of the TOE and client.

Step 2: The evaluator shall configure the TOE to communicate with the client using IEEE 802.11-2020 and a 256-bit (64 hex values 0-f) pre-shared key, setting up the connections as described in the operational guidance. The pre-shared key is only used for testing.

Step 3: The evaluator shall start the sniffing tool, initiate a connection between the TOE and client, and allow the client to authenticate, associate, and successfully complete the four-way handshake with the TOE.

Step 4: The evaluator shall set a timer for one minute, at the end of which the evaluator shall disconnect the TOE from the client and stop the sniffer.

Step 5: The evaluator shall identify the four-way handshake frames (denoted EAPOL-key in Wireshark captures) and derive the PTK and GTK from the four-way handshake frames and pre-shared key as specified in IEEE 802.11-2020.

Step 6: The evaluator shall select the first data frame from the captured packets that was sent between the TOE and client after the four-way handshake successfully completed, and with the frame control value 0x4208 (the first two bytes are 08 42). The evaluator shall use the GTK to decrypt the data portion of the selected packet as specified in IEEE 802.11-2020 and verify that the decrypted data contains ASCII-readable text.

Step 7: The evaluator shall repeat Step 6 for the next two data frames with frame control value 0x4208.

The evaluator shall also perform the following testing based on the supported GTK distribution methods:

AES Key Wrap (AES-KW Tests)

- *Test FCS_CKM.2/GTK:1: The evaluator shall test the authenticated encryption functionality of AES-KW for each combination of the following input parameter lengths:*
 - *128 and 256 bit key encryption keys (KEKs)*
 - *Three plaintext lengths:*
 1. *One of the plaintext lengths will be two semi-blocks (128 bits).*
 2. *One of the plaintext lengths will be three semi-blocks (192 bits).*
 3. *The third data unit length will be the longest supported plaintext length less than or equal to 64 semi-blocks (4096 bits).*

For each combination, generate a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KW authenticated encryption. To determine correctness, the evaluator shall use the same key and plaintext values and encrypt them using a known good implementation of AES-KW authenticated-encryption, and ensure that the resulting respective ciphertext values are identical.

- *Test FCS_CKM.2/GTK:2: The evaluator shall test the authenticated-decryption functionality of AES-KW using the same test as for authenticated-encryption, replacing plaintext values with ciphertext values and AES-KW authenticated-encryption with AES-KW authenticated-decryption. Additionally, the evaluator shall modify one byte of the ciphertext, attempt to decrypt the modified ciphertext, and ensure that a failure is returned rather than plaintext.*

AES Key Wrap with Padding (AES-KWP Tests)

- *Test FCS_CKM.2/GTK:3:*

The evaluator shall test the authenticated-encryption functionality of AES-KWP for each combination of the following input parameter lengths:

128 and 256 bit key encryption keys (KEKs)

Three plaintext lengths. One plaintext length will be one octet. One plaintext length will be 20 octets (160 bits). One plaintext length will be the longest supported plaintext length less than or equal to 512 octets (4096 bits).

Use a set of 100 key and plaintext pairs and obtain the ciphertext that results from AES-KWP authenticated encryption. To determine correctness, the evaluator shall use the AES-KWP authenticated-encryption function of a known good implementation.

- *Test FCS_CKM.2/GTK:4: The evaluator shall test the authenticated-decryption functionality of AES-KWP using the same test as for AES-KWP authenticated-encryption, replacing plaintext values with ciphertext values and AES-KWP authenticated-encryption with AES-KWP authenticated-decryption. Additionally, the evaluator shall modify one byte of the ciphertext, attempt to decrypt the modified ciphertext, and ensure that a failure is returned rather than plaintext.*

FCS_CKM.2/PMK Cryptographic Key Distribution (PMK)

This component must be included in the ST if the TOE implements any of the following features:

- [802.11bb Support](#)

FCS_CKM.2.1/PMK

The TSE shall **receive the 802.11 Pairwise Master Key (PMK)** in accordance with a specified cryptographic key distribution method: [*from 802.1X Authorization Server*] that meets the following: [*IEEE 802.11-2020*] **and does not expose the cryptographic keys.**

Application Note: This requirement applies to the Pairwise Master Key that is received from the RADIUS server by the TOE. The intent of this requirement is to ensure conformant TOEs implement 802.1X authentication prior to establishing secure communications with the client. The intent is that any LiFi AS product evaluated against this PP-Module will support both WPA2-Enterprise and WPA3-Enterprise at a minimum and certificate-based authentication mechanisms and therefore disallows implementations that support only pre-shared keys. Because communications with the RADIUS server are required to be performed over a protected connection, the transfer of the PMK will be protected.

Evaluation Activities ▼

[FCS_CKM.2/PMK](#)

TSS

The evaluator shall examine the TSS to determine that it describes how the PMK is transferred (that is, through what EAP attribute) to the TOE.

Guidance

There are no additional Guidance evaluation activities for this component.

Tests

The evaluator shall establish a session between the TOE and a RADIUS server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the RADIUS server and the TOE during a successful attempt to connect a wireless client to the TOE to determine that the PMK is not exposed.

FIA_PSK_EXT.1 Pre-Shared Key Composition

This component must be included in the ST if any of the following SFRs are included:

- [FCS_RADSEC_EXT.1](#)

FIA_PSK_EXT.1.1

The TSE shall be able to use pre-shared keys for [**selection:** RADIUS over TLS (RadSec), IPsec, WPA3-SAE, WPA3-SAE-PK, IEEE 802.11 WPA2-PSK, [**assignment:** other protocols that use pre-shared keys]].

FIA_PSK_EXT.1.2

The TSE shall be able to accept text-based pre-shared keys that:

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

FIA_PSK_EXT.1.3

The TSSF shall be able to [**selection:** *accept, generate using the random bit generator specified in FCS_RBG.1*] bit-based pre-shared keys.

Application Note: This requirement must be included if IPsec or another protocol that uses pre-shared keys is claimed, and pre-shared key authentication is selected (e.g., "Pre-shared Keys" is selected in FCS_IPSEC_EXT.1.13 or "pre-shared keys" is selected in FCS_RADSEC_EXT.1.2). The intent of this requirement is that all protocols will support both text-based and bit-based pre-shared keys.

For the length of the text-based pre-shared keys, a common length (22 characters) is required to help promote interoperability. If other lengths are supported, they should be listed in the assignment; this assignment can also specify a range of values (e.g., "lengths from 5 to 55 characters").

For FIA_PSK_EXT.1.3, the ST author specifies whether the TSSF merely accepts bit-based pre-shared keys or is capable of generating them. If it generates them, the requirement specifies that they must be generated using the RBG provided by the TOE.

Evaluation Activities ▼

FIA_PSK_EXT.1

TSS

The evaluator shall verify that the TSS describes

1. the protocols that can use pre-shared keys and that these are consistent with the selections made in [FIA_PSK_EXT.1.1](#).
2. the allowable values for pre-shared keys and that they are consistent with the selections made in [FIA_PSK_EXT.1.2](#).
3. the way bit-based pre-shared keys are procured and that it is consistent with the selections made in [FIA_PSK_EXT.1.3](#).

Guidance

The evaluator shall examine the operational guidance to determine that it provides guidance to administrators on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the range of lengths supported. The guidance must specify the allowable characters for pre-shared keys, and that list must be a superset of the list contained in [FIA_PSK_EXT.1.2](#).

The evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement or for generating a bit-based pre-shared key (or both).

Tests

The evaluator shall also perform the following tests for each protocol (or instantiation of a protocol, if performed by a different implementation on the TOE). Note that one or more of these tests can be

performed with a single test case.

- Test FIA_PSK_EXT.1:1: The evaluator shall compose a pre-shared key of 22 characters that contains a combination of the allowed characters in accordance with the operational guidance and demonstrates that a successful protocol negotiation can be performed with the key.
- Test FIA_PSK_EXT.1:2: [conditional]: If the TQE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length; a length inside the allowable range; and invalid lengths beyond the supported range (both higher and lower). The minimum, maximum, and included length tests should be successful, and the invalid lengths must be rejected by the TQE.
- Test FIA_PSK_EXT.1:3: [conditional]: If the TQE does not generate bit-based pre-shared keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.
- Test FIA_PSK_EXT.1:4: [conditional]: If the TQE does generate bit-based pre-shared keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.

FTP_ITC.1/Client Inter-TSF Trusted Channel (LiFi 802.11 Client Access)

This component must be included in the ST if the TQE implements any of the following features:

- **802.11bb Support**

FTP_ITC.1.1/Client

The TSF shall provide a communication channel **using WPA3-Enterprise, WPA2-Enterprise, and [selection: WPA3-SAE, WPA3-SAE-PK, WPA2-PSK, no other mode]** as defined by IEEE 802.11-2020 to provide a trusted channel between itself and **LiFi clients** that is logically distinct from other communication channels and provides assured identification of its endpoints and protection of the channel data from modification or disclosure.

Application Note: This requirement has been iterated from its definition in the NDcPP to mandate support for 802.11 specifically for LiFi client access.

FTP_ITC.1.2/Client

The TSF shall permit [**LiFi clients**] to initiate communication via the trusted channel.

FTP_ITC.1.3/Client

The TSF shall initiate communication via the trusted channel for [*no services*].

Evaluation Activities ▼

FTP_ITC.1/Client

This component is adequately evaluated when performing the evaluation activities for FTP_ITC.1 in the Network Device, version 4.0 Base-PP. The evaluator shall repeat the tests from that SER for

each claimed LiFi communications channel.

A.3.3 SFRs for Distributed TOE Implementations

FCS_CKM.2/DISTRIB Cryptographic Key Distribution (802.11 Keys)

This component must be included in the ST if the TOE implements any of the following features:

- *Distributed TOE*

FCS_CKM.2.1/DISTRIB

The TSF shall distribute **IEEE 802.11** keys in accordance with a specified cryptographic key distribution method [**assignment**: *trusted channel protocol specified in FPT_ITT.1 (from Base.PP)*] that meets the following: [**assignment**: *applicable standards as referenced by the distribution method specified in the Base.PP*].

Application Note: This SFR is required for use in distributed TOEs that support 802.11bb for distribution of keys to access nodes. G.vlc TOEs will not claim this as they are used for point-to-point communications and do not have a distributed architecture.

If claimed, this requirement applies to any key necessary for successful IEEE 802.11 connections not covered by [FCS_CKM.2/GTK](#). In cases where a key must be distributed to other APs, this communication must be performed via a mechanism of commensurate cryptographic strength. Because communications with any component of a distributed TOE are required to be performed over a trusted connection, the transfer of these keys will be protected.

Evaluation Activities ▼

[FCS_CKM.2/DISTRIB](#)

TSS

The evaluator shall examine the TSS to determine that it describes which keys are distributed outside the TOE, where they are sent, and the purpose for this transfer.

Guidance

If this function is dependent on TOE configuration, the evaluator shall confirm that the operational guidance contains instructions for how to configure that the keys are adequately protected.

Tests

This requirement will be tested in conjunction with the tests for the cryptographic primitives, the secure protocols, and FPT_ITT.1 (Base.PP).

A.3.4 SFRs for G.vlc Implementations

FTP_ITC.1/Gvlc Inter-TSF Trusted Channel (MACsec)

This component must be included in the ST if the TOE implements any of the following features:

- *G.vlc Support*

FTP_ITC.1.1/Gvlc

The TSE shall provide a communication channel **for G.vlc using MACsec to provide a trusted channel** between itself and a **MACsec peer** that is logically distinct from other communication channels and provides assured identification of its endpoints and protection of the channel data from modification or disclosure.

Application Note: This requirement has been iterated from its definition in the NDcPP to mandate support for MACsec.

FTP_ITC.1.2/Gvlc

The TSE shall permit [**selection: the TSE, a MACsec peer**] to initiate communication via the trusted channel.

FTP_ITC.1.3/Gvlc

The TSE shall initiate communication via the trusted channel for [*communications with MACsec peers that require the use of MACsec*].

Evaluation Activities ▼

FTP_ITC.1/Gvlc

G.vlc is implemented using MACsec. To demonstrate the implementation of G.vlc, the evaluator shall apply the test activities for FCS_MACSEC_EXT.1, FCS_MACSEC_EXT.2, FCS_MACSEC_EXT.3, and FCS_MACSEC_EXT.4 as defined in the PP-Module for MACsec Ethernet Encryption.

Appendix B - Selection-based Requirements

B.1 Auditable Events for Selection-based SFRs

Table 10: Auditable Events for Selection-based Requirements

Requirement	Auditable Events	Additional Audit Record Contents
FCS_RADSEC_EXT.1	No events specified	N/A
FCS_RADSEC_EXT.2	No events specified	N/A

B.2 Cryptographic Support (FCS)

FCS_RADSEC_EXT.1 RadSec

The inclusion of this selection-based component depends upon selection in [FTP_ITC.1.1/8021X](#).

FCS_RADSEC_EXT.1.1

The TSS shall implement RADIUS over TLS as specified in RFC 6614 to communicate securely with a RADIUS server.

FCS_RADSEC_EXT.1.2

The TSS shall perform peer authentication using [**selection:** *X.509v3 certificates, pre-shared keys*].

Application Note: This SFR is applicable if "RADIUS over TLS" is selected in [FTP_ITC.1/8021X](#).

If [X.509v3 certificates](#) is selected in [FCS_RADSEC_EXT.1.2](#), then [FCS_TLSC_EXT.1](#) from the Functional Package for TLS must be claimed and "mutual authentication" must be selected in [FCS_TLSC_EXT.1.1](#). If [pre-shared keys](#) is selected in [FCS_RADSEC_EXT.1.2](#), then [FCS_RADSEC_EXT.2](#) and [FIA_PSK_EXT.1](#) in this PP-Module must be claimed.

Evaluation Activities ▼

[FCS_RADSEC_EXT.1](#) TSS

The evaluator shall verify that the TSS description includes the use of RADIUS over TLS, as described in RFC 6614. If [X.509v3 certificates](#) is selected, the evaluator shall ensure that the TSS

description includes the use of client-side certificates for TLS mutual authentication.

Guidance

The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the guidance.

Tests

The evaluator shall demonstrate the ability to successfully establish a RADIUS over TLS connection with a RADIUS server. This test will be performed with X.509v3 certificates if selected and performed with pre-shared keys if selected.

FCS_RADSEC_EXT.2 RadSec Using Pre-Shared Keys

The inclusion of this selection-based component depends upon selection in [FCS_RADSEC_EXT.1.2](#).

FCS_RADSEC_EXT.2.1

The TSF shall implement **TLS 1.2** and [selection: TLS 1.3, no other TLS versions] when acting as a RADIUS over TLS client that supports the following ciphersuites:
[selection:

- TLS 1.2 ciphersuites:[selection:
 - TLS_ECDHE_PSK_WITH_AES_256_GCM_SHA384 as defined in RFC 8442
 - TLS_DHE_PSK_WITH_AES_256_GCM_SHA384 as defined in RFC 5487

]

- TLS 1.3 ciphersuites:
 - TLS_AES_256_GCM_SHA384 as defined in RFC 8446

[selection:

- TLS_AES_128_GCM_SHA256 as defined in RFC 8446
- [assignment: other TLS 1.3 ciphersuites]

]

].

Application Note: The above ciphersuites are only for use when the TSF is acting as a RADIUS over TLS client, not for other uses of the TLS protocol. The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported.

This requirement must be included if [pre-shared keys](#) is selected in [FCS_RADSEC_EXT.1.2](#).

FCS_RADSEC_EXT.2.2

The TSF shall be able to [selection: *accept, generate using the random bit generator specified in FCS_RBG.1*] bit-based pre-shared keys.

[*FCS_RADSEC_EXT.2*](#)

TSS

The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also verify that the TSS contains a description of the denial of old SSL and TLS versions. The evaluator shall examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG.1.

Guidance

The evaluator shall verify that any configuration necessary to meet the requirement must be contained in the guidance.

The evaluator shall also check the guidance documentation to ensure that it contains instructions on configuring the TOE so that RADIUS over TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements).

The evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys or generating a bit-based pre-shared key (or both).

Tests

*The testing for this requirement is addressed by the functional claims made in [*Functional Package for Transport Layer Security \(TLS\), version 2.1*](#) for this interface.*

Appendix C - Extended Component Definitions

This appendix contains the definitions for all extended requirements specified in the PP-Module.

C.1 Extended Components Table

All extended components specified in the PP-Module are listed in this table:

Table 11: Extended Component Definitions

Functional Class	Functional Components
Cryptographic Support (FCS)	FCS_RADSEC_EXT RadSec
Identification and Authentication (FIA)	FIA_8021X_EXT 802.1X Port Access Entity (Authenticator) Authentication FIA_PSK_EXT Pre-Shared Keys
Protection of the TSE (FPT)	FPT_ACF_EXT Access Controls FPT_ASLR_EXT Address Space Layout Randomization FPT_SBOP_EXT Stack Buffer Overflow Protection FPT_W^X_EXT Write XOR Execute Memory Pages
Security Management (FMT)	FMT_SMR_EXT Security Management Restrictions

C.2 Extended Component Definitions

C.2.1 Cryptographic Support (FCS)

This PP-Module defines the following extended components as part of the FCS class originally defined by CC Part 2:

C.2.1.1 FCS_RADSEC_EXT RadSec

Family Behavior

Components in this family describe requirements for implementation of the RadSec (RADIUS over TLS) protocol.

Component Leveling



[FCS_RADSEC_EXT.1](#), RadSec, requires the [TSF](#) to implement RadSec using a specified peer authentication method.

[FCS_RADSEC_EXT.2](#), RadSec Using Pre-Shared Keys, requires the [TSF](#) to implement RadSec using pre-shared key authentication in a manner that conforms to relevant [TLS](#) specifications.

Management: FCS_RADSEC_EXT.1

No specific management functions are identified.

Audit: FCS_RADSEC_EXT.1

There are no auditable events foreseen.

FCS_RADSEC_EXT.1 RadSec

Hierarchical to: No other components.

Dependencies to: [FCS_TLSC_EXT.1](#) [TLS](#) Client Protocol
[FIA_PSK_EXT.1](#) Pre-Shared Key Composition
[FIA_X509_EXT.1](#) X.509v3 Certificate Validation

FCS_RADSEC_EXT.1.1

The [TSF](#) shall implement [RADIUS](#) over [TLS](#) as specified in RFC 6614 to communicate securely with a [RADIUS](#) server.

FCS_RADSEC_EXT.1.2

The [TSF](#) shall perform peer authentication using [**assignment**: *some authentication method*].

Management: FCS_RADSEC_EXT.2

No specific management functions are identified.

Audit: FCS_RADSEC_EXT.2

There are no auditable events foreseen.

FCS_RADSEC_EXT.2 RadSec Using Pre-Shared Keys

Hierarchical to: No other components.

Dependencies to: [FCS_CKM.1](#) Cryptographic Key Generation
[FCS_COP.1](#) Cryptographic Operation
[FCS_RADSEC_EXT.1](#) RadSec
[FCS_RBG.1](#) Random Bit Generation

FCS_RADSEC_EXT.2.1

The [TSF](#) shall implement [**assignment**: *list of allowed TLS versions*] when acting as a [RADIUS](#) over [TLS](#) client that supports the following ciphersuites: [**assignment**: *list of supported ciphersuites*].

FCS_RADSEC_EXT.2.2

The TSF shall be able to [selection: *accept, generate using the random bit generator specified in FCS_RBG.1*] bit-based pre-shared keys.

C.2.2 Identification and Authentication (FIA)

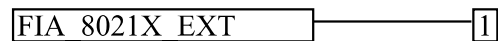
This PP-Module defines the following extended components as part of the FIA class originally defined by CC Part 2:

C.2.2.1 FIA_8021X_EXT 802.1X Port Access Entity (Authenticator) Authentication

Family Behavior

Components in this family describe requirements for implementation of 802.1X port-based network access control.

Component Leveling



[FIA_8021X_EXT.1](#), 802.1X Authentication, requires the TSF to securely implement IEEE 802.1X as an authenticator.

Management: FIA_8021X_EXT.1

No specific management functions are identified.

Audit: FIA_8021X_EXT.1

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

- Attempts to access the 802.1X controlled port prior to successful completion of the authentication exchange

FIA_8021X_EXT.1 802.1X Authentication

Hierarchical to: No other components.

Dependencies to: No dependencies

FIA_8021X_EXT.1.1

The TSF shall conform to IEEE Standard 802.1X for a Port Access Entity (PAE) in the “Authenticator” role.

FIA_8021X_EXT.1.2

The TSF shall support communications to a RADIUS authentication server conforming to RFCs 2865, 2866, and 3579.

FIA_8021X_EXT.1.3

The TSF shall ensure that no access to its 802.1X controlled port is given to the wireless client prior to successful completion of this authentication exchange.

C.2.2.2 FIA_PSK_EXT Pre-Shared Keys

Family Behavior

Components in this family describe requirements for using pre-shared keys.

Component Leveling



[FIA_PSK_EXT.1](#), Pre-Shared Key Composition, requires the [TSF](#) to support pre-shared keys that meet various characteristics for specific communications usage.

Management: FIA_PSK_EXT.1

No specific management functions are identified.

Audit: FIA_PSK_EXT.1

There are no auditable events foreseen.

FIA_PSK_EXT.1 Pre-Shared Key Composition

Hierarchical to: No other components.

Dependencies to: [FCS_RBG.1](#) Random Bit Generation

FIA_PSK_EXT.1.1

The [TSF](#) shall be able to use pre-shared keys for [**assignment:** *pre-shared key protocols*].

FIA_PSK_EXT.1.2

The [TSF](#) shall be able to accept text-based pre-shared keys that:

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

FIA_PSK_EXT.1.3

The [TSF](#) shall be able to [**selection:** *accept, generate using the random bit generator specified in [FCS_RBG.1](#)*] bit-based pre-shared keys.

C.2.3 Protection of the TSF (FPT)

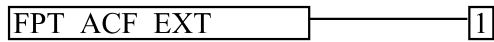
This [PP-Module](#) defines the following extended components as part of the FPT class originally defined by [CC](#) Part 2:

C.2.3.1 FPT_ACF_EXT Access Controls

Family Behavior

This family defines specific [TOE](#) components that are protected against unprivileged access. This is a new family defined for the FPT class.

Component Leveling



[FPT_ACF_EXT.1](#), Access Controls, requires the TSF to prohibit unauthorized users from reading or modifying specific TSF data.

Management: FPT_ACF_EXT.1

There are no management functions foreseen.

Audit: FPT_ACF_EXT.1

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

- Unauthorized attempts to perform operations against protected data

FPT_ACF_EXT.1 Access Controls

Hierarchical to: No other components.

Dependencies to: No dependencies.

FPT_ACF_EXT.1.1

The TSF shall implement access controls which prohibit unprivileged users from modifying:

- Kernel and its drivers and modules
- Security audit logs
- Shared libraries
- System executables
- System configuration files
- [**assignment:** *other objects*].

FPT_ACF_EXT.1.2

The TSF shall implement access controls which prohibit unprivileged users from reading:

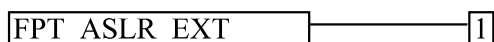
- Security audit logs
- System-wide credential repositories
- [**assignment:** *list of other objects*].

C.2.3.2 FPT_ASLR_EXT Address Space Layout Randomization

Family Behavior

This family defines the ability of the TOE to implement address space layout randomization (ASLR). This is a new family defined for the FPT class.

Component Leveling



[FPT_ASLR_EXT.1](#), Address Space Layout Randomization, defines the ability of the TOE to use ASLR as well as the objects that ASLR is applied to.

Management: FPT_AS LR_EXT.1

There are no management functions foreseen.

Audit: FPT_AS LR_EXT.1

There are no auditable events foreseen.

FPT_AS LR_EXT.1 Address Space Layout Randomization

Hierarchical to: No other components.

Dependencies to: No dependencies.

FPT_AS LR_EXT.1.1

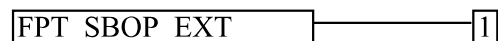
The `.TSF` shall always randomize process address space memory locations with [**selection:** 8, [*assignment:* number greater than 8]] bits of entropy except for [**assignment:** list of explicit exceptions].

C.2.3.3 FPT_SBOP_EXT Stack Buffer Overflow Protection

Family Behavior

This family requires the `.TSF` to be compiled using stack-based buffer overflow protections. This is a new family defined for the FPT class.

Component Leveling



[FPT_SBOP_EXT.1](#), Stack Buffer Overflow Protection, requires the `.TSF` to be compiled using stack-based buffer overflow protections or to store data in such a manner that a stack-based buffer overflow cannot compromise the `.TSF`.

Management: FPT_SBOP_EXT.1

There are no management activities foreseen.

Audit: FPT_SBOP_EXT.1

There are no auditable events foreseen.

FPT_SBOP_EXT.1 Stack Buffer Overflow Protection

Hierarchical to: No other components.

Dependencies to: No dependencies.

FPT_SBOP_EXT.1.1

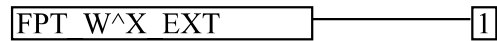
The `.TSF` shall [**selection:** *employ stack-based buffer overflow protections, not store parameters or variables in the same data structures as control flow values*].

C.2.3.4 FPT_W^X_EXT Write XOR Execute Memory Pages

Family Behavior

This family defines the ability of the T.O.E to implement data execution prevention (DEP) by preventing memory from being both writable and executable. This is a new family defined for the FPT class.

Component Leveling



FPT_W^X_EXT.1, Write XOR Execute Memory Pages, defines the ability of the T.O.E to prevent memory from being simultaneously writable and executable unless otherwise specified.

Management: FPT_W^X_EXT.1

There are no management functions foreseen.

Audit: FPT_W^X_EXT.1

There are no auditable events foreseen.

FPT_W^X_EXT.1 Write XOR Execute Memory Pages

Hierarchical to: No other components.

Dependencies to: No dependencies.

FPT_W^X_EXT.1.1

The T.S.F shall prevent allocation of any memory region with both write and execute permissions except for [assignment: *list of exceptions*].

C.2.4 Security Management (FMT)

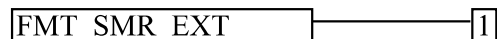
This PP-Module defines the following extended components as part of the FMT class originally defined by CC Part 2:

C.2.4.1 FMT_SMR_EXT Security Management Restrictions

Family Behavior

Components in this family describe architectural restrictions on security administration that are not defined in CC Part 2.

Component Leveling



FMT_SMR_EXT.1, No Administration from Client, requires the T.S.F to reject remote administration from a wireless client by default.

Management: FMT_SMR_EXT.1

No specific management functions are identified.

Audit: FMT_SMR_EXT.1

There are no auditable events foreseen.

FMT_SMR_EXT.1 No Administration from Client

Hierarchical to: No other components.

Dependencies to: FMT_SMF.1 Specification of Management Functions

FMT_SMR_EXT.1.1

The TSE shall ensure that the ability to remotely administer the TOE from a wireless client shall be disabled by default.

Appendix D - Acronyms

Table 12: Acronyms

Acronym	Meaning
AES	Advanced Encryption Standard
AP	Access Point
ASLR	Address Space Layout Randomization
Base-PP	Base Protection Profile
CBC	Cipher Block Chaining
CC	Common Criteria
CCM	Counter Mode with CBC-Message Authentication Code
CCMP	CCM mode Protocol
CEM	Common Evaluation Methodology
cPP	Collaborative Protection Profile
CTR	Counter (encryption mode)
EAP	Extensible Authentication Protocol
GCM	Galois-Counter Mode
GTK	Group Temporal Key
IPsec	Internet Protocol Security
MAC	Media Access Control
NDcPP	Network Device collaborative Protection Profile
OE	Operational Environment
PAE	Port Access Entity
PMK	Pairwise Master Key
PP	Protection Profile
PP-Configuration	Protection Profile Configuration
PP-Module	Protection Profile Module

PTK	Pairwise Transient Key
RADIUS	Remote Authentication Dial In User Service
RBG	Random Bit Generator
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SSID	Service Set Identifier
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSEI	TSF Interface
TSS	TOE Summary Specification
WPA	Wi-Fi Protected Access

Appendix E - Bibliography

Table 13: Bibliography

Identifier	Title
[NDcPP]	collaborative Protection Profile for Network Devices, Version 4.0, December 22, 2025
[CC]	Common Criteria for Information Technology Security Evaluation - • Part 1: Introduction and general model, CCMB-2022-11-001, CC:2022, Revision 1, November 2022. • Part 2: Security functional requirements, CCMB-2022-11-002, CC:2022, Revision 1, November 2022. • Part 3: Security assurance requirements, CCMB-2022-11-003, CC:2022, Revision 1, November 2022. • Part 4: Framework for the specification of evaluation methods and activities, CCMB-2022-11-004, CC:2022, Revision 1, November 2022. • Part 5: Pre-defined packages of security requirements, CCMB-2022-11-005, CC:2022, Revision 1, November 2022.
[CEM]	Common Methodology for Information Technology Security Evaluation - • Evaluation methodology, CCMB-2022-11-006, CC:2022, Revision 1, November 2022.