Apple Inc.

## Apple macOS 14 Sonoma

# Security Target

Version 1.7

December 19, 2025

Prepared by:


www.lightshipsec.com

# Document History

| Version | Date | Author | Description |
|---|---|---|---|
| 1.0 | 27 Feb 2025 | G. NICKEL | Release for Check In |
| 1.1 | 2-May-2025 | G. NICKEL<br>G. MCLEARN | Address ECR Comments |
| 1.2 | 13-May-2025 | G. MCLEARN | Evaluator comments |
| 1.3 | 11-June-2025 | G. MCLEARN | Removed unsupported platforms, other minor changes related to testing. |
| 1.4 | 26-Aug-2025 | G. MCLEARN | Updated auditing claims to include Apple Unified Logging. Other minor changes related to testing. Adding new TDs. |
| 1.5 | 21-Nov-2025 | G. MCLEARN | Addressing evaluator comments |
| 1.6 | 5-Dec-2025 | G. MCLEARN | Addressing evaluator comments |
| 1.7 | 19-Dec-2025 | G. MCLEARN | Addressing ECR |

## Trademarks

Apple's trademarks applicable to this document are listed in https://www.apple.com/legal/intellectual-property/trademark/appletmlist.html

Other company, product, and service names may be trademarks or service marks of others.

## Copyright

# Table of Contents

# List of Tables

# 1 Introduction

## 1.1 Overview

1 This Security Target (ST) defines the Apple macOS 14 Sonoma Target of Evaluation (TOE) on Apple Silicon for the purposes of Common Criteria (CC) evaluation.

2 Apple macOS 14 Sonoma is a general-purpose operating system that supports a computing environment for multiple users, applications, and includes Bluetooth and Wi-fi client functionality.

3 The macOS 14 Sonoma software is a Unix-based graphical operating system. The macOS core is a POSIX compliant operating system built on top of the XNU kernel with standard Unix facilities available from the command line interface.

## 1.2 Identification

**Table 1: Evaluation identifiers**

| | |
|---|---|
| Target of Evaluation | Apple macOS 14 Sonoma<br>Build: 14.8.2 |
| Security Target | Apple macOS 14 Sonoma Security Target, v1.7, December 19, 2025 |

## 1.3 Conformance Claims

4 This ST supports the following conformance claims:

a) CC version 3.1 Release 5

b) CC Part 2 extended

c) CC Part 3 extended

d) PP-Configuration for General Purpose Operating System, Bluetooth, and Wireless Local Area Network Clients (CFG_GPOS_BT_WLANC_V1.0).
This PP-Configuration includes the following components:

i) Base-PP: Protection Profile for General Purpose Operating Systems, Version 4.3 (PP_OS_V4.3);[1]

ii) PP-Module: PP-Module for Bluetooth, Version 1.0 (MOD_BT_V1.0); and

[1] The PP-Configuration uses the PP short term of “PP_GPOS_V4.3”, whereas the officially published Protection Profile for General Purpose Operating Systems Version 4.3 is designated by NIAP as “PP_OS_V4.3”. This ST uses the officially published short term while acknowledging that the Base-PP is unambiguously identified in the PP-Configuration by its correct full name and version number.

iii) PP-Module: PP-Module for WLAN Clients, Version 1.0 (MOD_WLANC_V1.0).

e) Functional Package for Transport Layer Security (TLS), Version 1.1 (PKG_TLS_V1.1)

f) NIAP Technical Decisions per Table 2

**Table 2: NIAP Technical Decisions**

| TD # | Name | Source | Applicability Rationale |
|---|---|---|---|
| TD0442 | Updated TLS Ciphersuites for TLS Package | PKG_TLS_V1.1 | Applicable. |
| TD0469 | Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1 | PKG_TLS_V1.1 | Not Applicable - FCS_TLSS_EXT.1 not claimed. |
| TD0499 | Testing with pinned certificates | PKG_TLS_V1.1 | Applicable. |
| TD0513 | CA Certificate loading | PKG_TLS_V1.1 | Applicable. |
| TD0600 | Conformance claim sections updated to allow for MOD_VPNC_V2.3 | MOD_BT_V1.0 | Applicable. |
| TD0640 | Handling BT devices that do not support encryption | MOD_BT_V1.0 | Applicable. |
| TD0645 | Bluetooth audit details | MOD_BT_V1.0 | Applicable. |
| TD0650 | Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4 | MOD_BT_V1.0 | Applicable. |
| TD0667 | Move Set Wireless Freq Band to Optional/Objective | MOD_WLANC_V1.0 | Applicable. |
| TD0671 | Bluetooth PP-Module updated to allow for new PP and PP-Module Versions | MOD_BT_V1.0 | Applicable. |
| TD0675 | Make FPT_W^X_EXT.1 Optional | PP_OS_V4.3 | Applicable. |
| TD0685 | BT missing multiple SFR-to-Obj mappings | MOD_BT_V1.0 | Applicable. |

| TD # | Name | Source | Applicability Rationale |
|---|---|---|---|
| TD0691 | OSPP 4.3 Conditional authentication testing | PP_OS_V4.3 | Applicable. |
| TD0693 | Typos in OSPP 4.3 | PP_OS_V4.3 | Applicable. |
| TD0696 | Removal of 160 bit selection from FCS_COP.1/HASH & FCS_COP.1/KEYHMAC | PP_OS_V4.3 | Applicable. |
| TD0701 | Incomplete selection reference in FCS_CKM_EXT.4 TSS activities | PP_OS_V4.3 | Applicable. |
| TD0703 | Removal of FIA_X509_EXT.2/WLAN evaluation activities for revocation checking | MOD_WLANC_V1.0 | Applicable. |
| TD0707 | Formatting corrections for MOD_BT | MOD_BT_V1.0 | Applicable. |
| TD0710 | WPA version restrictions | MOD_WLANC_V1.0 | Applicable. |
| TD0712 | Support for Bluetooth Standard 5.3 | PP_OS_V4.3 | Applicable. |
| TD0713 | Functional Package SFR mappings to objectives | PP_OS_V4.3 | Applicable. |
| TD0726 | Corrections to (D)TLSS SFRs in TLS 1.1 FP | PKG_TLS_V1.1 | Not Applicable – FCS_DTLSS_EXT.1 or FCS_TLSS_EXT.1 not claimed. |
| TD0739 | PKG_TLS_V1.1 has 2 different publication dates | PKG_TLS_V1.1 | Not Applicable – Does not impact this Security Target. |
| TD0770 | TLSS.2 connection with no client cert | PKG_TLS_V1.1 | Not Applicable – FCS_TLSS_EXT.2 not claimed. |

| TD # | Name | Source | Applicability Rationale |
|---|---|---|---|
| TD0773 | Updates to FIA_X509_EXT.1 for Exception Processing and Test Conditions | PP_OS_V4.3 | Applicable. |
| TD0779 | Updated Session Resumption Support in TLS package V1.1 | PKG_TLS_V1.1 | Not Applicable – FCS_TLSS_EXT.1 not claimed. |
| TD0789 | Correction to TLS Selection in FIA_X509_EXT.2.1 | PP_OS_V4.3 | Applicable. |
| TD0797 | Addition of FCS_WPA_EXT to ECD | MOD_WLANC_V1.0 | Applicable. |
| TD0812 | Updated CC Conformance Claims in OSPP | PP_OS_V4.3 | Applicable. |
| TD0821 | Corrections to ECD for PP_OS_V4.3 | PP_OS_V4.3 | Applicable. |
| TD0837 | Updates to WLAN Client PP-Module allow-lists | MOD_WLANC_V1.0 | Applicable. |
| TD0839 | Clarification for Local Administration in FTP_TRP.1.3 | PP_OS_V4.3 | Applicable. |
| TD0844 | Addition of Assurance Package for Flaw Remediation V1.0 Conformance Claim | PP_OS_V4.3 | Not Applicable – Does not impact this Security Target. |
| TD0904 | Addition of MOD_VPNC_V2.5 to Conformance Claims | PP_OS_V4.3 | Not Applicable – Does not impact this Security Target. |
| TD0906 | Clarification to List of Examples in FPT_SBOP_EXT.1 | PP_OS_V4.3 | Applicable. |
| TD0914 | Addition of PKG_TLS_V2.0 to Conformance Claims | PP_OS_V4.3 | Not Applicable – TD is optional and not applied to this evaluation. |

| TD # | Name | Source | Applicability Rationale |
|---|---|---|---|
| TD0920 | Clarification for FMT_SMF.1/WLAN Table 3 | MOD_WLANC_V1.0 | Applicable |
| TD0930 | Clarification when CTR_DRBG is Selected for FCS_RBG_EXT.1.2 in PP_OS_V4.3 | PP_OS_V4.3 | Applicable. |
| TD0955 | Adding FIPS 186-5 in PP_OS_V4.3 | PP_OS_V4.3 | Applicable. |
| TD0958 | Correction to Referenced PPs in FTP_ITC_EXT.1 | PP_OS_V4.3 | Applicable. |

## 1.4 Terminology

**Table 3: Terminology**

| Term | Definition |
|---|---|
| AES | Advanced Encryption Standard |
| ABM | Apple Business Manager |
| ACE | Access Control Entry |
| AES | Advanced Encryption Standard |
| AP | Access Point |
| APFS | Apple File System |
| API | Application Programming Interface |
| app | Application |
| ASLR | Address Space Layout Randomization |
| BD_ADDR | Bluetooth Device Address |
| BR/EDR | Basic Rate/Enhanced Data Rate |
| BSD | Berkeley Software Distribution |

| Term | Definition |
|---|---|
| BSM | Basic Security Module |
| CA | Certificate Authority |
| CAVP | Cryptographic Algorithm Validation Program |
| CC | Common Criteria |
| CCM | Counter with CBC-MAC |
| CEM | Common Evaluation Methodology |
| CIFS | Common Internet File System |
| CMC | Certificate Management over CMS |
| CMS | Cryptographic Message Syntax |
| CSP | Critical Security Parameters |
| CTR | Counter (a mode of AES) |
| CVE | Common Vulnerabilities and Exposures |
| DAR | Data At Rest |
| DEK | Data Encryption Key |
| DEP | Data Execution Prevention |
| DNS | Domain Name System |
| DRBG | Deterministic Random Bit Generator |
| DSS | Digital Signature Standard |
| ECC | Elliptic Curve Cryptography |
| ECDH | Elliptic Curve Diffie-Hellman |
| ECDHE | ECDH Ephemeral |
| EKU | extendedKeyUsage |
| EST | Enrollment over Secure Transport |

| Term | Definition |
|---|---|
| FIPS | Federal Information Processing Standards |
| GCM | Galois/Counter Mode |
| GID | Group Identifier |
| GPOS | General Purpose Operating System |
| HCI | Host Controller Interface |
| HMAC | Hash-based Message Authentication Code |
| ID | Identifier or Identity |
| IP | Internet Protocol |
| KAS | Key Agreement Scheme |
| KEK | Key Encryption Key |
| L2CAP | Logical Link Control and Adaptation Protocol |
| LE | Low Energy |
| LLB | Low-Level Bootloader |
| MAC | Message Authentication Code |
| NIAP | National Information Assurance Partnership |
| NIST | National Institute of Standards and Technology |
| OCSP | Online Certificate Status Protocol |
| OID | Object Identifier |
| OS | Operating System |
| PBKDF | Password-Based Key Derivation Function |
| PGP | Pretty Good Privacy |
| PII | Personally Identifiable Information |
| PIN | Personal Identification Number |

| Term | Definition |
|---|---|
| PKI | Public Key Infrastructure |
| POSIX | Portable Operating System Interface |
| PP | Protection Profile |
| RA | Registration Authority |
| RBG | Random Bit Generator |
| RFCOMM | Radio Frequency Communication |
| ROM | Read Only Memory |
| RSA | Rivest-Shamir-Adleman |
| SAN | Subject Alternative Name |
| SAR | Security Assurance Requirement |
| SCEP | Simple Certificate Enrollment Protocol |
| SEE | Separate Execution Environment |
| SEP | Secure Enclave Processor |
| SFR | Security Functional Requirement |
| SHA | Secure Hash Algorithm |
| SL1 | Security Level 1 (FIPS 140-3) |
| SMB | Server Message Block |
| SoC | System on a Chip |
| SSP | Secure Simple Pairing |
| ST | Security Target |
| TLS | Transport Layer Security |
| TOE | Target of Evaluation |
| TRNG | True Random Number Generator |

| Term | Definition |
|---|---|
| TSF | TOE Security Function |
| TSFI | TSF Interface |
| TSS | TOE Summary Specification |
| UDID | Unique Device Identifier |
| UID | User Identifier |
| UUID | Universally Unique Identifier |
| WAP | Wireless Access Point |
| WLAN | Wireless Local Area Network |
| WLANC | Wireless Local Area Network Client |
| XNU | X is Not Unix |

# 2 TOE Description

## 2.1 Type

5 The TOE is a general-purpose operating system (GPOS) for Apple Mac computers containing Apple Silicon series processors and provides wireless LAN and Bluetooth functionality.

### 2.1.1 Architecture

6 The TOE includes the macOS and sepOS software that is pre-installed on the Apple Silicon Macs identified in section 2.4.2. Each device is protected by the Apple Secure Enclave.

7 The Secure Enclave is a dedicated secure subsystem integrated into Apple Systems on Chip (SoCs). Utilizing a dedicated processor (Secure Enclave Processor or SEP), it is isolated from the main processor to provide an extra layer of security designed to keep sensitive user data secure, even when the Application Processor kernel becomes compromised. The SEP runs the sepOS firmware on a separate, dedicated boot ROM. It is based on a customized version of the L4 microkernel.

8 The Secure Enclave supports the TOE for secure boot, and the generation of secure random data used in cryptographic key generation for AES encryption/decryption.

9 The executing TOE is divided into user space and kernel space. User space contains processes that execute in their own protected memory space and access services provided by the kernel. Kernel space contains the macOS kernel (including device

drivers and kernel extensions) that also executes in its own protected memory space. The kernel enforces process separation, provides processes with controlled access to hardware devices, and implements many other OS features. The SEP is only accessible by the macOS kernel.

10 Apple Silicon devices include a chip that implements Bluetooth and wireless LAN functionality. The chip model depends on the hardware platforms listed in section 2.4.2.

## 2.2 Usage

11 The following are expected use cases for the TOE per their respective Protection Profiles:

a) **PP_OS_V4.3:** USE CASE 1. The TOE provides a platform for end user device such as desktops, laptops, convertibles, and tablets.

b) **MOD_BT_V1.0:** USE CASE 4. The Bluetooth functionality provided by the TOE is part of the general-purpose operating system itself. No standalone third-party applications are necessary to be installed.

c) **MOD_WLANC_V1.0.** USE CASE 1. The TOE provides a wireless LAN client (WLANC) that is part of the general purpose operating system.

## 2.3 Logical Scope

12 The TOE provides the following security functions:

a) **Audit Data Generation.** The TOE generates audit events for all start-up and shut-down functions, and all auditable events as specified by the conformance claims defined in 1.3. Audit events are generated for the following audit functions:

i) Start-up and shut-down of the audit functions,

ii) Authentication events (Success/Failure),

iii) Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes),

iv) Privilege or role escalation events (Success/Failure),

v) Administrator or root-level access events (Success/Failure),

vi) Events related to Bluetooth connections and user authorizations,

vii) Events related to connections to Wi-Fi networks and associated user authentications,

viii) Attempts to load and revoke X.509 certificates; and,

ix) Execution of TSF-self tests.

Each audit record contains the date and time of the event, type of event, subject identity (if applicable), and outcome (success or failure) of the event.

b) **User Data Protection.** The TOE implements access controls that prevent unprivileged users from accessing files and directories owned by other users.

c) **Identification and Authentication.** All users must be authenticated to the TOE prior to carrying out any management actions. The TOE supports:

   i) password-based authentication,

   ii) authentication based on username and a PIN that releases asymmetric key stored in OE-protected storage.

   The TOE will lock out user accounts after a defined number of unsuccessful authentication attempts have been met. The TOE supports Bluetooth Secure Simple Pairing (SSP). It requires user authorization and mutual authentication during pairing. It also discards pairing attempts and session initialization from Bluetooth devices to which an active session pre-exists. The TOE requires explicit user authorization when pairing with an untrusted device.

   The TOE also supports the use of X.509 certificates for the purpose of identifying itself and/or its users for TLS connections as well as for connecting to Wi-Fi networks using EAP-TLS.

d) **Security Management.** The TOE can perform management functions. The administrator has full access to carry out all management functions; whereas the user will have limited privileges.

e) **Protection of the TSF.** The TOE implements the following protection of TSF data functions:

   i) Access Controls.

   ii) Address space layout randomization (ASLR) with 16 bits of entropy.

   iii) Stack buffer overflow protection.

   iv) Verification of integrity of the boot-chain and operating system executable code.

   v) Trusted software updates using digital signatures.

f) **Trusted Path/Channels.** The TOE supports TLS v1.2 for trusted channel communications. The TOE uses TLS to securely communicate with the Apple Update Server. The TOE enforces encryption when transmitting data over Bluetooth for both BR/EDR and LE and terminates the connection if the connected device stops encrypting. The TOE enforces 802.1X EAP-TLS authentication to wireless access points and protects wireless traffic using encryption when connecting via the wireless LAN client.

g) **TOE Access.** Before establishing a user session, the TOE will display an advisory warning message regarding unauthorized use of the OS.

h) **Cryptographic Support.** The TOE includes the Apple corecrypto v14.0 cryptographic libraries and is supported by the onboard Apple SEP Hardware for performing user space, kernel space, and SEP cryptographic operations. In addition, it uses a software and hardware noise source for entropy generation. The TOE implements TLS 1.2 for secure communications with remote servers.

The Bluetooth hardware implements the AES-CCM-128 cryptographic functionality used when connecting to Bluetooth devices. The TOE implements WPA3 to secure 802.11 wireless traffic protected using AES-GCMP-256 cryptographic algorithms. Relevant Cryptographic Algorithm Validation Program (CAVP) certificates are shown in Appendix 7.1.

## 2.4 Physical Scope

13 The physical boundary of the TOE is the installation image which includes both macOS and sepOS installed on Apple devices. If required, the Administrator can leverage the built-in update feature to check for and install the same version of the TOE or newer version. Additionally, the TOE can be manually installed using the built-in "App Store".

14 The Apple devices covered by this evaluation are listed in section 2.4.2.

### 2.4.1 Non-TOE Components

15 The following components must be present in the operational environment to operate the TOE in the evaluated configuration:

**Table 4: TOE Operational Environment**

| Component | Description |
|---|---|
| Hardware platform | See section 2.4.2 |
| Apple Update Server | Server that allows the TOE to download updates |
| Smart Card,<br>Smart Card Reader | An encrypted card that allows the TOE to authenticate a user when used with smart card reader devices. |
| NTP Server | Server that allows the TOE to synchronise its time |
| Wireless Access Point | Access point for the TOE to connect to for WLAN connectivity capable of mediating 802.1X authentication. |
| RADIUS server | A RADIUS server capable of providing 802.1X services. |

### 2.4.2 Hardware

16 The evaluated configuration includes the following Apple devices running on Apple Silicon:

**Table 5: Hardware Platforms – Apple Silicon**

| Marketing Name | Model | Model Identifier | Processor (Micro Architecture) | Security Chip | BT Version (BT/WiFi Chip) |
|---|---|---|---|---|---|
| 2023 | | | | | |
| MacBook Pro (16-inch, 2023) | A2780 | Mac14,6 | M2 Max (ARMv8.6-A) | SEP v2.0 | 5.3 (4388) |
| | | Mac14,10 | M2 Pro (ARMv8.6-A) | | |
| MacBook Pro (14-inch, 2023) | A2779 | Mac14,5 | M2 Max (ARMv8.6-A) | SEP v2.0 | 5.3 (4388) |
| | | Mac14,9 | M2 Pro (ARMv8.6-A) | | |
| Mac mini (M2 Pro, 2023) | A2816 | Mac14,12 | M2 Pro (ARMv8.6-A) | SEP v2.0 | 5.3 (4388) |
| Mac mini (M2, 2023) | A2686 | Mac14,3 | M2 (ARMv8.6-A) | SEP v2.0 | 5.3 (4388) |
| 2022 | | | | | |
| MacBook Pro (13-inch, M2, 2022) | A2338 | Mac14,7 | M2 (ARMv8.6-A) | SEP v2.0 | 5.0 (4378) |
| MacBook Air (M2, 2022) | A2861 | Mac14,2 | M2 (ARMv8.6-A) | SEP v2.0 | 5.0 (4387) |
| Mac Studio (2022) | A2615 | Mac13,2 | M1 Ultra (ARMv8.5-A) | SEP v2.0 | 5.0 (4387) |
| | | Mac13,1 | M1 Max (ARMv8.5-A) | | |
| 2021 | | | | | |
| MacBook Pro (16-inch, 2021) | A2485 | MacBookPro18,2 | M1 Max (ARMv8.5-A) | SEP v2.0 | 5.0 (4387) |
| | | MacBookPro18,1 | M1 Pro (ARMv8.5-A) | | |
| MacBook Pro (14-inch, 2021) | A2442 | MacBookPro18,4 | M1 Max (ARMv8.5-A) | | 5.0 (4387) |

| Marketing Name | Model | Model Identifier | Processor (Micro Architecture) | Security Chip | BT Version (BT/WiFi Chip) |
|---|---|---|---|---|---|
| | | MacBookPro18,3 | M1 Pro (ARMv8.5-A) | SEP v2.0 | |
| iMac (24-inch, M1, 2021) | A2438 | iMac21,1 | M1 (ARMv8.5-A) | SEP v2.0 | 5.0 (4378) |
| | A2439 | iMac21,2 | | | |
| 2020 | | | | | |
| Mac mini (M1, 2020) | A2348 | Macmini9,1 | M1 (ARMv8.5-A) | SEP v2.0 | 5.0 (4378) |
| MacBook Air (M1, 2020) | A2337 | MacBookAir10,1 | M1 (ARMv8.5-A) | SEP v2.0 | 5.0 (4378) |
| MacBook Pro (13-inch, M1, 2020) | A2338 | MacBookPro17,1 | M1 (ARMv8.5-A) | SEP v2.0 | 5.0 (4378) |

### 2.4.3 Guidance Documents

17 The TOE includes the following guidance documents in PDF format which are publicly available on the NIAP Product Compliance List (PCL) alongside this Security Target:

a) [CCGUIDE] Apple macOS 14 Sonoma Common Criteria Guide, Version 1.2, December 19, 2025

# 3 Security Problem Definition

## 3.1 Threats

**Table 6: Threats**

| Identifier | Description | PP Origin |
|---|---|---|
| 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 OS with the intent of compromise. Engagement may consist of altering existing legitimate communications. | PP_OS_V4.3, MOD_BT_V1.0 |
| T.NETWORK_EAVESDROP | An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between applications and services that are running on or part of the OS. | PP_OS_V4.3, MOD_BT_V1.0 |
| T.LOCAL_ATTACK | An attacker may compromise applications running on the OS. The compromised application may provide maliciously formatted input to the OS through a variety of channels including unprivileged system calls and messaging via the file system. | PP_OS_V4.3 |
| T.LIMITED_PHYSICAL_ACCESS | An attacker may attempt to access data on the OS while having a limited amount of time with the physical device. | PP_OS_V4.3 |
| T.TSF_FAILURE | Security mechanisms of the TOE generally build up from a primitive set of mechanisms (e.g., memory management, privileged modes of process execution) to more complex sets of mechanisms. Failure of the primitive mechanisms could lead to a compromise in more complex mechanisms, resulting in a compromise of the TSF. | MOD_WLANC_V1.0 |

| Identifier | Description | PP Origin |
|---|---|---|
| T.UNAUTHORIZED_ACCESS | A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data. | MOD_WLANC_V1.0 |
| T.UNDETECTED_ACTIONS | Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated. | MOD_WLANC_V1.0 |

## 3.2 Assumptions

**Table 7: Assumptions**

| Identifier | Description | PP Origin |
|---|---|---|
| A.PLATFORM | The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of this PP. | PP_OS_V4.3 |
| A.PROPER_USER | The user of the OS is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. At the same time, malicious software could act as the user, so requirements which confine malicious subjects are still in scope. | PP_OS_V4.3 |
| A.PROPER_ADMIN | The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy. | PP_OS_V4.3 |
| A.NO_TOE_BYPASS | Information cannot flow between the wireless client and the internal wired network without passing through the TOE. | MOD_WLANC_V1.0 |
| A.TRUSTED_ADMIN | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. | MOD_WLANC_V1.0 |

## 3.3 Organizational Security Policies

18 No organizational security policies have been identified in PP_OS_V4.3, MOD_BT_V1.0, or MOD_WLANC_V1.0.

## 3.4 Security Objectives for the TOE

**Table 8: Security Objectives**

| Identifier | Description | PP Origin |
|---|---|---|
| O.ACCOUNTABILITY | Conformant OSes ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and | PP_OS_V4.3 |

| Identifier | Description | PP Origin |
|---|---|---|
| | discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise. | |
| O.INTEGRITY | Conformant OSes ensure the integrity of their update packages. OSes are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSes provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. | PP_OS_V4.3 |
| O.MANAGEMENT | To facilitate management by users and the enterprise, conformant OSes provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration and application execution control. | PP_OS_V4.3 |
| O.PROTECTED_STORAGE | To address the issue of loss of confidentiality of credentials in the event of loss of physical control of the storage medium, conformant OSes provide data-at-rest protection for credentials. Conformant OSes also provide access controls which allow users to keep their files private from other users of the same system. | PP_OS_V4.3 |
| O.PROTECTED_COMMS | To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSes provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed outside of the platform. | PP_OS_V4.3, MOD_BT_V1.0 |

| Identifier | Description | PP Origin |
|---|---|---|
| O.AUTH_COMM | The TOE will provide a means to ensure that it is communicating with an authorized access point and not some other entity pretending to be an authorized access point, and will provide assurance to the access point of its identity. | MOD_WLANC_V1.0 |
| O.CRYPTOGRAPHIC_FUNCTIONS | The TOE will provide or use cryptographic functions (i.e., encryption/decryption and digital signature operations) to maintain the confidentiality and allow for detection of modification of data that are transmitted outside the TOE and its host environment. | MOD_WLANC_V1.0 |
| O.SELF_TEST | The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly. | MOD_WLANC_V1.0 |
| O.SYSTEM_MONITORING | The TOE will provide the capability to generate audit data. | MOD_WLANC_V1.0 |
| O.TOE_ADMINISTRATION | The TOE will provide mechanisms to allow administrators to be able to configure the TOE. | MOD_WLANC_V1.0 |
| O.WIRELESS_ACCESS_POINT_CONNECTION | The TOE will provide the capability to restrict the wireless access points to which it will connect. | MOD_WLANC_V1.0 |

## 3.5 Security Objectives for the Operational Environment

**Table 9: Operational environment objectives**

| Identifier | Description | PP Origin |
|---|---|---|
| OE.PLATFORM | The OS relies on being installed on trusted hardware. | PP_OS_V4.3 |

| Identifier | Description | PP Origin |
|---|---|---|
| OE.PROPER_USER | The user of the OS is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy. Standard user accounts are provisioned in accordance with the least privilege model. Users requiring higher levels of access should have a separate account dedicated for that use. | PP_OS_V4.3 |
| OE.PROPER_ADMIN | The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy. | PP_OS_V4.3 |
| OE.NO_TOE_BYPASS | Information cannot flow between external and internal networks located in different enclaves without passing through the TOE. | MOD_WLANC_V1.0 |
| OE.TRUSTED_ADMIN | TOE administrators are trusted to follow and apply all administrator guidance in a trusted manner. | MOD_WLANC_V1.0 |

# 4 Security Requirements

## 4.1 Conventions

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

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

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

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

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

e) **Iteration.** Indicated by appending an additional identifier e.g. “FCS_COP.1/SKC” or “FCS_COP.1(1)” depending on the PP/MOD convention.

20 **Note:** Operations performed within the Security Target are denoted within brackets []. Operations shown without brackets are reproduced from PP_OS_V4.3, MOD_BT_V1.0, and PKG_TLS_V1.1.

## 4.2 Extended Components Definition

21 All extended components (identified by SFRs appended with 'EXT' in Table 10 below) are reproduced directly from the PP indicated in the Source column. Refer to PP_OS_V4.3 Appendix C, MOD_BT_V1.0 Appendix C, and MOD_WLANC_V1.0 Appendix C for complete extended components definitions.

22 PP_OS_V4.3 implements the following extended security assurance requirements (SAR) that are referenced in Table 17:

a) ALC_TSU_EXT.1 Timely Security Updates

Refer to PP_OS_V4.3 section 5.2 for more information.

## 4.3 Functional Requirements

**Table 10: Summary of SFRs**

| Requirement | Title | Type | Source |
|---|---|---|---|
| FAU_GEN.1 | Audit Data Generation (Refined) | Mandatory | PP_OS_V4.3 |
| FAU_GEN.1/BT | Audit Data Generation (Bluetooth) | Mandatory | MOD_BT_V1.0 |
| FAU_GEN.1/WLAN | Audit Data Generation (Wireless LAN) | Mandatory | MOD_WLANC_V1.0 |
| FCS_CKM.1 | Cryptographic Key Generation (Refined) | Mandatory | PP_OS_V4.3 |
| FCS_CKM.1/WPA | Cryptographic Key Generation (Symmetric Keys for WPA2/WPA3 Connections) | Mandatory | MOD_WLANC_V1.0 |
| FCS_CKM.2 | Cryptographic Key Establishment (Refined) | Mandatory | PP_OS_V4.3 |
| FCS_CKM.2/WLAN | Cryptographic Key Distribution (Group Temporal Key for WLAN) | Mandatory | MOD_WLANC_V1.0 |
| FCS_CKM_EXT.4 | Cryptographic Key Destruction | Mandatory | PP_OS_V4.3 |
| FCS_COP.1/ENCRYPT | Cryptographic Operation - Encryption/Decryption (Refined) | Mandatory | PP_OS_V4.3 |
| FCS_COP.1/HASH | Cryptographic Operation - Hashing (Refined) | Mandatory | PP_OS_V4.3 |

| Requirement | Title | Type | Source |
|---|---|---|---|
| FCS_COP.1/SIGN | Cryptographic Operation - Signing (Refined) | Mandatory | PP_OS_V4.3 |
| FCS_COP.1/ KEYHMAC | Cryptographic Operation - Keyed-Hash Message Authentication (Refined) | Mandatory | PP_OS_V4.3 |
| FCS_RBG_EXT.1 | Random Bit Generation | Mandatory | PP_OS_V4.3 |
| FCS_STO_EXT.1 | Storage of Sensitive Data | Mandatory | PP_OS_V4.3 |
| FCS_CKM_EXT.8 | Bluetooth Key Generation | Mandatory | MOD_BT_V1.0 |
| FCS_TLS_EXT.1 | TLS Protocol | Mandatory | PKG_TLS_V1.1 |
| FCS_TLSC_EXT.1 | TLS Client Protocol | Selection | PKG_TLS_V1.1 |
| FCS_TLSC_EXT.1/ WLAN | TLS Client Protocol (EAP-TLS for WLAN) | Mandatory | MOD_WLANC_V1.0 |
| FCS_TLSC_EXT.2 | TLS Client Support for Mutual Authentication | Selection | PKG_TLS_V1.1 |
| FCS_TLSC_EXT.2/ WLAN | TLS Client Support for Supported Groups Extension (EAP-TLS for WLAN) | Selection | MOD_WLANC_V1.0 |
| FCS_TLSC_EXT.5 | TLS Client Support for Supported Groups Extension | Selection | PKG_TLS_V1.1 |
| FCS_WPA_EXT.1 | Supported WPA Versions | Mandatory | MOD_WLANC_V1.0 |
| FDP_ACF_EXT.1 | Access Controls for Protecting User Data | Mandatory | PP_OS_V4.3 |
| FIA_AFL.1 | Authentication failure handling (Refined) | Mandatory | PP_OS_V4.3 |
| FIA_UAU.5 | Multiple Authentication Mechanisms (Refined) | Mandatory | PP_OS_V4.3 |
| FIA_X509_EXT.1 | X.509 Certificate Validation | Mandatory | PP_OS_V4.3 |
| FIA_X509_EXT.1/ WLAN | X.509 Certificate Validation | Mandatory | MOD_WLANC_V1.0 |

| Requirement | Title | Type | Source |
|---|---|---|---|
| FIA_X509_EXT.2 | X.509 Certificate Authentication | Mandatory | PP_OS_V4.3 |
| FIA_X509_EXT.2/ WLAN | X.509 Certificate Authentication (EAP-TLS for WLAN) | Mandatory | MOD_WLANC_V1.0 |
| FIA_X509_EXT.6 | X.509 Certificate Storage and Management | Mandatory | MOD_WLANC_V1.0 |
| FIA_BLT_EXT.1 | Bluetooth User Authorization | Mandatory | MOD_BT_V1.0 |
| FIA_BLT_EXT.2 | Bluetooth Mutual Authentication | Mandatory | MOD_BT_V1.0 |
| FIA_BLT_EXT.3 | Rejection of Duplicate Bluetooth Connections | Mandatory | MOD_BT_V1.0 |
| FIA_BLT_EXT.4 | Secure Simple Pairing | Mandatory | MOD_BT_V1.0 |
| FIA_BLT_EXT.6 | Trusted Bluetooth Device User Authorization | Mandatory | MOD_BT_V1.0 |
| FIA_BLT_EXT.7 | Untrusted Bluetooth Device User Authorization | Mandatory | MOD_BT_V1.0 |
| FIA_PAE_EXT.1 | Port Access Entity Authentication | Mandatory | MOD_WLANC_V1.0 |
| FMT_MOF_EXT.1 | Management of security functions behavior | Mandatory | PP_OS_V4.3 |
| FMT_SMF_EXT.1 | Specification of Management Functions | Mandatory | PP_OS_V4.3 |
| FMT_MOF_EXT.1/BT | Management of Security Functions Behavior | Mandatory | MOD_BT_V1.0 |
| FMT_SMF_EXT.1/BT | Specification of Management Functions | Mandatory | MOD_BT_V1.0 |
| FMT_SMF.1/WLAN | Specification of Management Functions (WLAN Client) | Mandatory | MOD_WLANC_V1.0 |
| FPT_ACF_EXT.1 | Access controls | Mandatory | PP_OS_V4.3 |
| FPT_ASLR_EXT.1 | Address Space Layout Randomization | Mandatory | PP_OS_V4.3 |
| FPT_SBOP_EXT.1 | Stack Buffer Overflow Protection | Mandatory | PP_OS_V4.3 |

| Requirement | Title | Type | Source |
|---|---|---|---|
| FPT_TST_EXT.1 | Boot Integrity | Mandatory | PP_OS_V4.3 |
| FPT_TST_EXT.3/WLAN | TSF Cryptographic Functionality Testing (WLAN Client) | Mandatory | MOD_WLANC_V1.0 |
| FPT_TUD_EXT.1 | Trusted Update | Mandatory | PP_OS_V4.3 |
| FPT_TUD_EXT.2 | Trusted Update for Application Software | Mandatory | PP_OS_V4.3 |
| FTA_TAB.1 | Default TOE access banners | Optional | PP_OS_V4.3 |
| FTA_WSE_EXT.1 | Wireless Network Access | Mandatory | MOD_WLANC_V1.0 |
| FTP_ITC_EXT.1 | Trusted channel communication | Mandatory | PP_OS_V4.3 |
| FTP_ITC.1/WLAN | Trusted Channel Communication (Wireless LAN) | Mandatory | MOD_WLANC_V1.0 |
| FTP_TRP.1 | Trusted Path | Mandatory | PP_OS_V4.3 |
| FTP_BLT_EXT.1 | Bluetooth Encryption | Mandatory | MOD_BT_V1.0 |
| FTP_BLT_EXT.2 | Persistence of Bluetooth Encryption | Mandatory | MOD_BT_V1.0 |
| FTP_BLT_EXT.3/BR | Bluetooth Encryption Parameters (BR/EDR) | Mandatory | MOD_BT_V1.0 |
| FTP_BLT_EXT.3/LE | Bluetooth Encryption Parameters (LE) | Selection | MOD_BT_V1.0 |

### 4.3.1 Security Audit (FAU)

#### FAU_GEN.1 Audit Data Generation (Refined)

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

a) Start-up and shut-down of the audit functions;

b) All auditable events for the [*not specified*] level of audit; and [

c)

- *Authentication events (Success/Failure);*

- *Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes);*
- *Privilege or role escalation events (Success/Failure);*
- *[*
  - Administrator or root-level access events (Success/Failure)

  *]*

].

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*none*]

### FAU_GEN.1/BT Audit Data Generation (Bluetooth)

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

a) Start-up and shutdown of the audit functions

b) All auditable events for the [*not specified*] level of audit

c) [*Specifically defined auditable events in the Auditable Events table*].

**Table 11: Auditable Events (MOD_BT_V1.0)**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_CKM_EXT.8 | None. | |
| FIA_BLT_EXT.1 | Failed user authorization of Bluetooth device. | User authorization decision (e.g., user rejected connection, incorrect pin entry). |
| | Failed user authorization for local Bluetooth Service. | [complete] BD_ADDR and [no other information]. |
| | | Bluetooth profile. Identity of local service with [service ID]. |
| FIA_BLT_EXT.2 | Initiation of Bluetooth connection. | [complete] BD_ADDR and [no other information]. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| | Failure of Bluetooth connection. | Reason for failure. |
| FIA_BLT_EXT.4 | None. | |
| FIA_BLT_EXT.6 | None. | |
| FIA_BLT_EXT.7 | None. | |
| FTP_BLT_EXT.1 | None. | |
| FTP_BLT_EXT.2 | None. | |
| FTP_BLT_EXT.3/BR | None. | |
| FTP_BLT_EXT.3/LE (if claimed) | None. | |

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

a) Date and time of the event

b) Type of event

c) Subject identity

d) The outcome (success or failure) of the event

e) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*Additional information in the Auditable Events table*].

**Application Note:** This SFR was altered by TD0707. Table 11 was altered by TD0645.

### FAU_GEN.1/WLAN Audit Data Generation (Wireless LAN)

FAU_GEN.1.1/WLAN The TSF shall **[implement functionality]** to generate an audit record of the following auditable events:

a) Startup and shutdown of the audit functions;

b) All auditable events for [*not specified*] level of audit; and

c) [*all auditable events for mandatory SFRs specified in ~~Table 2~~* **Table 12** *and selected SFRs in ~~Table 5~~* **Table 13**].

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

a) Date and time of the event, type of event, subject identity, (if relevant) 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-**Module**/ST, [*Additional Audit Record Contents as specified in ~~Table 2~~* **Table 12** *and ~~Table 5~~* **Table 13**].

**Table 12: Auditable Events for Mandatory Requirements (MOD_WLANC_V1.0)**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FAU_GEN.1/WLAN | No events specified. | N/A |
| FCS_CKM.1/WPA | No events specified. | N/A |
| FCS_CKM.2/WLAN | No events specified. | N/A |
| FCS_TLSC_EXT.1/WLAN | Failure to establish an EAP-TLS session. | Reason for failure.<br>Non-TOE endpoint of connection. |
| FCS_TLSC_EXT.1/WLAN | Establishment/termination of an EAP-TLS session. | Non-TOE endpoint of connection. |
| FCS_WPA_EXT.1 | No events specified. | N/A |
| FIA_PAE_EXT.1 | No events specified. | N/A |
| FIA_X509_EXT.1/WLAN | Failure to validate X.509v3 certificate. | Reason for failure of validation. |
| FIA_X509_EXT.2/WLAN | No events specified. | N/A |
| FIA_X509_EXT.6 | Attempts to load certificates. | None. |
| FIA_X509_EXT.6 | Attempts to revoke certificates. | None. |
| FMT_SMF.1/WLAN | No events specified. | N/A |
| FPT_TST_EXT.3/WLAN | Execution of this set of TSF self-tests. | None. |
| FPT_TST_EXT.3/WLAN | [None]. | [None]. |

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FTA_WSE_EXT.1 | All attempts to connect to access points. | For each access point record the [Complete SSID and MAC] of the MAC Address<br>Success and failures (including reason for failure). |
| FTP_ITC.1/WLAN | All attempts to establish a trusted channel. | Identification of the non-TOE endpoint of the channel. |

**Table 13: Auditable Events for Selection-based Requirements (MOD_WLANC_V1.0)**

| Requirement | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_TLSC_EXT.2/WLAN | No events specified. | N/A |

### 4.3.2 Cryptographic Support (FCS)

**FCS_CKM.1** **Cryptographic Key Generation (Refined)**

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

- RSA schemes using cryptographic key sizes of 3072-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3
- ECC schemes using "NIST curves" P-384 and [P-256, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4

].

**Application Note:** This SFR was altered by TD0712 and TD0955.

**FCS_CKM.1/WPA** **Cryptographic Key Generation (Symmetric Keys for WPA2/WPA3 Connections)**

FCS_CKM.1.1/WPA The TSF shall generate **symmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [***PRF-384 and*** **[PRF-704] (as defined in IEEE 802.11- 2012)**] and specified key sizes [***256 bits and*** **[no other key sizes]**] **using a Random Bit Generator as specified in FCS_RBG_EXT.1.**

## FCS_CKM.2 Cryptographic Key Establishment (Refined)

FCS_CKM.2.1 The OS **shall implement functionality to perform cryptographic key establishment** in accordance with a specified cryptographic key **establishment** method: [

- RSA-based key establishment schemes that meets the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2"
- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"

].

## FCS_CKM.2/WLAN Cryptographic Key Distribution (Group Temporal Key for WLAN)

FCS_CKM.2.1/WLAN The TSF shall **decrypt Group Temporal Key** in accordance with a specified cryptographic key distribution method [*AES Key Wrap (as defined in RFC 3394) in an EAPOL-Key frame (as defined in IEEE 802.11-2012 for the packet format and timing considerations*] **and does not expose the cryptographic keys**.

## FCS_CKM_EXT.4 Cryptographic Key Destruction

FCS_CKM_EXT.4.1 The OS shall destroy cryptographic keys and key material in accordance with a specified cryptographic key destruction method [

- For volatile memory, the destruction shall be executed by a [
  - single overwrite consisting of [zeroes]

  ]
- For non-volatile memory that consists of [
  - destruction of all key encrypting keys (KEKs) protecting the target key according to FCS_CKM_EXT.4.1, where none of the KEKs protecting the target key are derived
  - the invocation of an interface provided by the underlying platform that

    [
    - instructs the underlying platform to destroy the abstraction that represents the key

    ]

  ]

] .

FCS_CKM_EXT.4.2 The OS shall destroy all keys and key material when no longer needed.

## FCS_COP.1/ENCRYPT Cryptographic Operation - Encryption/Decryption (Refined)

FCS_COP.1.1/ENCRYPT The **OS** shall perform [*encryption/decryption services for data*] in accordance with a specified cryptographic algorithm **[**

- AES-CTR (as defined in NIST SP 800-38A)

**] and [**

- AES Key Wrap (KW) (as defined in NIST SP 800-38F)
- AES-GCMP-256 (as defined in NIST SP 800-38D and IEEE 802.11ac-2013)
- AES-CCM (as defined in NIST SP 800-38C)
- AES-GCM (as defined in NIST SP 800-38D)

] and cryptographic key sizes 256-bit and [128-bit] ~~that meet the following: [assignment: list of standards]~~.

**Application Note:** This SFR was altered by TD0712.

## FCS_COP.1/HASH Cryptographic Operation – Hashing (Refined)

FCS_COP.1.1/HASH The **OS** shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm **[**

- SHA-256
- SHA-384
- SHA-512

**] and message digest sizes [**

- 256 bits
- 384 bits
- 512 bits

**]** that meet the following: [*FIPS Pub 180-4*].

**Application Note:** This SFR was altered by TD0696.

## FCS_COP.1/SIGN Cryptographic Operation – Signing (Refined)

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

- RSA schemes using cryptographic key sizes of [3072-bit or greater] that meet the following: FIPS PUB 186-4 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 4
- ECDSA schemes using "NIST curves" P-384 and [P-521] that meet the following: FIPS PUB 186-4 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5, SP 800-186 Section 3

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

**Application Note:** This SFR was altered by TD0955.

### FCS_COP.1/KEYHMAC Cryptographic Operation - Keyed-Hash Message Authentication (Refined)

FCS_COP.1.1/KEYHMAC The **OS** shall perform [*keyed-hash message authentication services*] in accordance with a specified cryptographic algorithm [SHA-256, SHA-384] with key sizes [***256 bits, 384 bits***] **and message digest** sizes [256 bits, 384 bits] that meet the following: [*FIPS Pub 198-1 The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard*].

**Application Note:** This SFR was altered by TD0696.

### FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The OS shall perform all deterministic random bit generation (DRBG) services in accordance with NIST Special Publication 800-90A using [

- CTR_DRBG (AES)

] .

FCS_RBG_EXT.1.2 The deterministic RBG used by the OS shall be seeded by an entropy source that accumulates entropy from a [

- software-based noise source
- platform-based noise source

] with a minimum of 256 bits of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

### FCS_STO_EXT.1 Storage of Sensitive Data

FCS_STO_EXT.1.1 The OS shall implement functionality to encrypt sensitive data stored in non-volatile storage and provide interfaces to applications to invoke this functionality.

### FCS_CKM_EXT.8 Bluetooth Key Generation

FCS_CKM_EXT.8.1 The TSF shall generate public/private ECDH key pairs every [*new connection attempt*].

### FCS_ TLS_EXT.1 TLS Protocol

FCS_TLS_EXT.1.1 The product shall implement [

- TLS as a client

].

### FCS_ TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1 The product shall implement TLS 1.2 (RFC 5246) and [no earlier TLS versions] as a client that supports the cipher suites [

- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] and also supports functionality for [

- mutual authentication

].

FCS_TLSC_EXT.1.2 The product shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.1.3 The product shall not establish a trusted channel if the server certificate is invalid [

- with no exceptions.

].

**Application Note:** This SFR has been modified by TD0442.

### FCS_ TLSC_EXT.1/WLAN TLS Client Protocol (EAP-TLS for WLAN)

FCS_TLSC_EXT.1.1/WLAN The TSF shall implement TLS 1.2 (RFC 5246) and [no other TLS version] in support of the EAP-TLS protocol as specified in RFC 5216 supporting the following cipher suites: [

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,

].

| | |
|---|---|
| FCS_TLSC_EXT.1.2/WLAN | The TSF shall generate random values used in the EAP-TLS exchange using the RBG specified in FCS_RBG_EXT.1. |
| FCS_TLSC_EXT.1.3/WLAN | The TSF shall use X509 v3 certificates as specified in FIA_X509_EXT.1/**WLAN**. |
| FCS_TLSC_EXT.1.4/WLAN | The TSF shall verify that the server certificate presented includes the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field. |
| FCS_TLSC_EXT.1.5/WLAN | The TSF shall allow an authorized administrator to configure the list of CAs that are allowed to sign authentication server certificates that are accepted by the TOE. |

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

| | |
|---|---|
| FCS_TLSC_EXT.2.1 | The product shall support mutual authentication using X.509v3 certificates. |

### FCS_TLSC_EXT.2/WLAN TLS Client Support for Supported Groups Extension (EAP-TLS for WLAN)

| | |
|---|---|
| FCS_TLSC_EXT.2.1/WLAN | The TSF shall present the Supported Groups extension in the Client Hello with the following NIST curves: [secp384r1]. |

### FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension

| | |
|---|---|
| FCS_TLSC_EXT.5.1 | The product shall present the Supported Groups Extension in the Client Hello with the supported groups [<br>• secp384r1,<br>• secp521r1,<br>]. |

### FCS_WPA_EXT.1 Supported WPA Versions

| | |
|---|---|
| FCS_WPA_EXT.1.1 | The TSF shall support WPA3 and [no other] security type. |

## 4.3.3 User Data Protection (FDP)

### FDP_ACF_EXT.1 Access Controls for Protecting User Data

FDP_ACF_EXT.1.1 The OS shall implement access controls which can prohibit unprivileged users from accessing files and directories owned by other users.

### 4.3.4 Identification and Authentication (FIA)

#### FIA_AFL.1 Authentication failure handling (Refined)

FIA_AFL.1.1 The **OS** shall detect when [

- an administrator configurable positive integer within [*1-50*]

] unsuccessful authentication attempts occur related to **events with** [

- **authentication based on user name and password**
- **authentication based on user name and a PIN that releases an asymmetric key stored in OE-protected storage**

] .

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts for an account has been **met**, the **OS** shall: **[Account Lockout]** .

#### FIA_UAU.5 Multiple Authentication Mechanisms (Refined)

FIA_UAU.5.1 The **OS** shall provide the following authentication mechanisms [

- **authentication based on username and password**
- **authentication based on username and a PIN that releases an asymmetric key stored in OE-protected storage**

] to support user authentication.

FIA_UAU.5.2 The OS shall authenticate any user's claimed identity according to the [*Authentication based on username and a password/PIN that release a set of keys stored in the TOE to unwrap locally stored files*].

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

FIA_X509_EXT.1.1 The OS shall implement functionality to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a trusted CA certificate
- The OS shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.

- The TSF shall validate that any CA certificate includes "Certificate Signing" as a purpose the key usage field
- The OS shall validate the revocation status of the certificate using [OCSP as specified in RFC 6960] with [no exceptions]
- The OS shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.
    - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing Purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.
    - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field. (conditional)

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

## FIA_X509_EXT.1/WLAN X.509 Certificate Validation

FIA_X509_EXT.1.1/WLAN The TSF shall validate certificates for **EAP-TLS** in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation
- The certificate path must terminate with a certificate in the Trust Anchor Database
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates

- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.

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

### FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The OS shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [TLS] connections.

**Application Note:** This SFR was altered by TD0789.

### FIA_X509_EXT.2/WLAN X.509 Certificate Authentication (EAP-TLS for WLAN)

FIA_X509_EXT.2.1/WLAN The TSF shall use X.509v3 certificates as defined by RFC 5280 to support [*[authentication for EAP-TLS exchanges]*].

### FIA_X509_EXT.6 X.509 Certificate Storage and Management

FIA_X509_EXT.6.1 The TSF shall [store and protect] certificate(s) from unauthorized deletion and modification.

FIA_X509_EXT.6.2 The TSF shall [provide the capability for authorized administrators to load X.509v3 certificates into the TOE] for use by the TSF.

### FIA_BLT_EXT.1 Bluetooth User Authorization

FIA_BLT_EXT.1.1 The TSF shall require explicit user authorization before pairing with a remote Bluetooth device.

### FIA_BLT_EXT.2 Bluetooth Mutual Authentication

FIA_BLT_EXT.2.1 The TSF shall require Bluetooth mutual authentication between devices prior to any data transfer over the Bluetooth link.

### FIA_BLT_EXT.3 Rejection of Duplicate Bluetooth Connections

FIA_BLT_EXT.3.1 The TSF shall discard pairing and session initialization attempts from a Bluetooth device address (BD_ADDR) to which an active session already exists.

### FIA_BLT_EXT.4 Secure Simple Pairing

FIA_BLT_EXT.4.1 The TOE shall support Bluetooth Secure Simple Pairing, both in the host and the controller.

FIA_BLT_EXT.4.2 The TOE shall support Secure Simple Pairing during the pairing process.

### FIA_BLT_EXT.6 Trusted Bluetooth Device User Authorization

FIA_BLT_EXT.6.1 The TSF shall require explicit user authorization before granting trusted remote devices access to services associated with the following Bluetooth profiles: [*none*].

### FIA_BLT_EXT.7 Untrusted Bluetooth Device User Authorization

FIA_BLT_EXT.7.1 The TSF shall require explicit user authorization before granting untrusted remote devices access to services associated with the following Bluetooth profiles: [*none*].

### FIA_PAE_EXT.1 Port Access Entity Authentication

FIA_PAE_EXT.1.1 The TSF shall conform to IEEE Standard 802.1X for a Port Access Entity (PAE) in the “Supplicant” role.

## 4.3.5 Security Management (FMT)

### FMT_MOF_EXT.1 Management of security functions behavior

FMT_MOF_EXT.1.1 The OS shall restrict the ability to perform the function indicated in the "Administrator" column in FMT_SMF_EXT.1.1 to the administrator.

### FMT_SMF_EXT.1 Specification of Management Functions

FMT_SMF_EXT.1.1 The OS shall be capable of performing the following management functions:

**Table 14: TOE Management Functions**

| # | Management Function | Administrator | User |
|---|---|---|---|
| 1 | Enable/disable [screen lock] | M | M |
| 2 | Configure [screen lock] inactivity timeout | M | M |
| 3 | import keys/secrets into the secure key storage | M | - |
| 4 | Configure local audit storage capacity | M | - |
| 5 | Configure minimum password length | M | - |
| 6 | Configure minimum number of special characters in password | M | - |
| 7 | Configure minimum number of numeric characters in password | - | - |
| 8 | Configure minimum number of uppercase characters in password | - | - |
| 9 | Configure minimum number of lowercase characters in password | - | - |
| 10 | Configure lockout policy for unsuccessful authentication attempts through [timeouts between attempts, limiting number of attempts during a time period] | - | - |
| 11 | Configure host-based firewall | M | - |
| 12 | Configure name/address of directory server with which to bind | - | - |
| 13 | Configure name/address of remote management server from which to receive management settings | - | - |
| 14 | Configure name/address of audit/logging server to which to send audit/logging records | - | - |
| 15 | Configure audit rules | M | - |
| 16 | Configure name/address of network time server | M | - |
| 17 | Enable/disable automatic software update | M | - |
| 18 | Configure Wi-Fi interface | M | - |

| # | Management Function | Administrator | User |
|---|---|---|---|
| 19 | Enable/disable Bluetooth interface | M | - |
| 20 | Enable/disable [*no other external interfaces*] | - | - |
| 21 | [*No other management functions to be provided by the TSF*] | - | - |

**Application Note:** This SFR was modified by TD0693.

**Application Note:** (M) Supported by the specified role.
( - ) Not supported by the specified role.
The use of 'M' and '-' as indicator markers are specifically required to be used as defined by TD0693. This is unlike other FMT_SMF_EXT.1 iterations from other modules defined in this section which use different status markers to indicate support. Functionally, an 'M' shown in this table has the same meaning as an 'X' shown in other iterations for the specific user roles.

**Application Note:** Regarding #18 ("Configure Wi-Fi interface"), the ability to configure the Wi-Fi interface is dependent on the specifics described in FMT_SMF.1/WLAN In the TOE, these options are managed through Configuration Profiles, which can only be managed by the administrator.

**Application Note:** Regarding #19 ("Enable/disable Bluetooth interface"), the ability to enable and disable the Bluetooth interface is dependent on the specifics described in FMT_SMF_EXT.1/BT. Within that SFR, the TOE is required to be able to enable and disable both Discoverable (for BR/EDR) and Advertising (for LE) modes. Only the administrator can perform both of these actions, hence, the selection in this table.

## FMT_MOF_EXT.1/BT Management of Security Functions Behavior

FMT_MOF_EXT.1.1/BT The OS shall restrict the ability to perform the function indicated in the "Administrator" column in FMT_SMF_EXT.1.1/**BT** to the administrator.

## FMT_SMF_EXT.1/BT Specification of Management Functions

FMT_SMF_EXT.1.1/BT The OS shall be capable of performing the following **Bluetooth** management functions:

**Table 15: Management functions (MOD_BT_V1.0)**

| Function | Administrator | User |
|---|---|---|
| BT-1. Configure the Bluetooth trusted channel. | X | - |

| Function | Administrator | User |
|---|---|---|
| • Disable/enable the Discoverable (for BR/EDR) and Advertising (for LE) modes; | | |
| BT-2. Change the Bluetooth device name (separately for BR/EDR and LE); | - | - |
| BT-3. Provide separate controls for turning the BR/EDR and LE radios on and off; | - | - |
| BT-4. Allow/disallow the following additional wireless technologies to be used with Bluetooth: [WiFi]; | - | - |
| BT-5. Configure allowable methods of Out of Band pairing (for BR/EDR and LE); | - | - |
| BT-6. Disable/enable the Discoverable (for BR/EDR) and Advertising (for LE) modes separately; | - | - |
| BT-7. Disable/enable the Connectable mode (for BR/EDR and LE); | - | - |
| BT-8. Disable/enable the Bluetooth [*assignment: list of Bluetooth service and/or profiles available on the OS (for BR/EDR and LE)*]; | - | - |
| BT-9. Specify minimum level of security for each pairing (for BR/EDR and LE); | - | - |

**Application Note:** (X) Supported by the specified role.
( - ) Not supported by the specified role.

## FMT_SMF.1/WLAN Specification of Management Functions (WLAN Client)

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

**Table 16: Management functions (MOD_WLANC_V1.0)**

| # | Management Function | Impl | Admin | User |
|---|---|---|---|---|
| WL-1 | configure security policy for each wireless network:<br>• [specify the CA(s) from which the TSF will accept WLAN authentication server certificate(s)],<br>• security type, | M | X | - |

| # | Management Function | Impl | Admin | User |
|---|---|---|---|---|
| | • authentication protocol,<br>• client credentials to be used for authentication, | | | |
| WL-2 | specify wireless networks (SSIDs) to which the TSF may connect | M | X | - |
| WL-3 | enable/disable disable wireless network bridging capability (for example, bridging a connection between the WLAN and cellular radios to function as a hotspot) authenticated by [passcode] | M | X | - |
| WL-4 | enable/disable certificate revocation list checking | O | - | - |
| WL-5 | disable ad hoc wireless client-to-client connection capability | O | X | - |
| WL-6 | disable roaming capability | O | - | - |
| WL-7 | enable/disable IEEE 802.1X pre-authentication | O | - | - |
| WL-8 | loading X.509 certificates into the TOE | O | X | - |
| WL-9 | revoke X.509 certificates loaded into the TOE | O | X | - |
| WL-10 | enable/disable and configure PMK caching:<br>• set the amount of time (in minutes) for which PMK entries are cached,<br>• set the maximum number of PMK entries that can be cached | O | - | - |
| WL-11 | configure security policy for each wireless network: set wireless frequency band to [2.4 GHz, 5 GHz, 6 GHz] | O | - | - |

**Application Note:** The "Impl" column is replicated directly from the module and indicates that the TOE must implement the functionality when there is an "M" status marker in the cell. When the column contains an 'O' status marker in the cell, this means that this ST can, at its discretion, implement the function. The specific implementation in this ST indicates a claimed function in the 'Admin' and/or 'User' roles with an 'X' (supported function) or '-' (unsupported function). Optional functions that are not claimed by the TOE for a specific role are indicated with a 'not supported' status marker and grey shading.

**Application Note:** This SFR has been modified by TD0667.

### 4.3.6 Protection of the TSF (FPT)

#### FPT_ACF_EXT.1 Access controls

FPT_ACF_EXT.1.1 The OS shall implement access controls which prohibit unprivileged users from modifying:

- Kernel and its drivers/modules
- Security audit logs
- Shared libraries
- System executables
- System configuration files
- [*TSF-data*
- *Applications*]

.

FPT_ACF_EXT.1.2 The OS shall implement access controls which prohibit unprivileged users from reading:

- Security audit logs
- System-wide credential repositories
- [*no other objects*].

#### FPT_ASLR_EXT.1 Address Space Layout Randomization

FPT_ASLR_EXT.1.1 The OS shall always randomize process address space memory locations with [ [ *16* ] ] bits of entropy except for [*no exception*].

#### FPT_SBOP_EXT.1 Stack Buffer Overflow Protection

FPT_SBOP_EXT.1.1 The OS shall [employ stack-based buffer overflow protections].

#### FPT_TST_EXT.1 Boot Integrity

FPT_TST_EXT.1.1 The OS shall verify the integrity of the bootchain up through the OS kernel and [

- no other executable code

] prior to its execution through the use of [

- a digital signature using a hardware-protected asymmetric key

] .

### FPT_TST_EXT.3/WLAN TSF Cryptographic Functionality Testing (WLAN Client)

FPT_TST_EXT.3.1/WLAN The [TOE] shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

FPT_TST_EXT.3.2/WLAN The [TOE] shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic services.

### FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The OS shall provide the ability to check for updates to the OS software itself and shall use a digital signature scheme specified in FCS_COP.1/SIGN to validate the authenticity of the response.

FPT_TUD_EXT.1.2 The OS shall [cryptographically verify] updates to itself using a digital signature prior to installation using schemes specified in FCS_COP.1/SIGN.

### FPT_TUD_EXT.2 Trusted Update for Application Software

FPT_TUD_EXT.2.1 The OS shall provide the ability to check for updates to application software and shall use a digital signature scheme specified in FCS_COP.1/SIGN to validate the authenticity of the response.

FPT_TUD_EXT.2.2 The OS shall cryptographically verify the integrity of updates to applications using a digital signature specified by FCS_COP.1/SIGN prior to installation.

## 4.3.7 TOE Access (FTA)

### FTA_TAB.1 Default TOE access banners

FTA_TAB.1.1 Before establishing a user session, the OS shall display an advisory warning message regarding unauthorized use of the OS.

### FTA_WSE_EXT.1 Wireless Network Access

FTA_WSE_EXT.1.1 The TSF shall be able to attempt connections only to wireless networks specified as acceptable networks as configured by the administrator in FMT_SMF.1.1/WLAN.

## 4.3.8 Trusted Path/Channel (FTP)

### FTP_ITC_EXT.1 Trusted channel communication

FTP_ITC_EXT.1.1 The OS shall use [

- TLS as conforming to the Functional Package for Transport Layer Security (TLS) as a [client]

]

and [

- no other protocols

]

] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: [*[update server]* ] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

**Application Note:** This SFR was altered by TD0789 and TD0958.

### FTP_ITC.1/WLAN Trusted Channel Communication (Wireless LAN)

FTP_ITC.1.1/WLAN The TSF shall use **802.11-2012, 802.1X, and EAP-TLS** to provide a **trusted** communication channel between itself and **a wireless access point** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2/WLAN The TSF shall permit [*the TSF*] to initiate communication via the trusted channel.

FTP_ITC.1.3/WLAN The TSF shall initiate communication via the trusted channel for [*wireless access point connections*].

### FTP_TRP.1 Trusted Path

FTP_TRP.1.1 The **OS** shall provide a communication path between itself and [local] users that is logically distinct from other communication paths and provides assured identification of its endpoints and protection of the communicated data from [modification, disclosure].

FTP_TRP.1.2 The **OS** shall permit [local users] to initiate communication via the trusted path.

FTP_TRP.1.3 The **OS** shall require use of the trusted path for [initial user authentication].

**Application Note:** This SFR has been modified by TD0839.

### FTP_BLT_EXT.1 Bluetooth Encryption

| | |
|---|---|
| FTP_BLT_EXT.1.1 | The TSF shall enforce the use of encryption when transmitting data over the Bluetooth trusted channel for BR/EDR and [LE]. |
| FTP_BLT_EXT.1.2 | The TSF shall use key pairs per FCS_CKM_EXT.8 for Bluetooth encryption. |

### FTP_BLT_EXT.2 Persistence of Bluetooth Encryption

| | |
|---|---|
| FTP_BLT_EXT.2.1 | The TSF shall [terminate the connection] if the remote device stops encryption while connected to the TOE. |

### FTP_BLT_EXT.3/BR Bluetooth Encryption Parameters (BR/EDR)

| | |
|---|---|
| FTP_BLT_EXT.3.1/BR | The TSF shall set the minimum encryption key size to [*128 bits*] for [*BR/EDR*] and not negotiate encryption key sizes smaller than the minimum size. |

### FTP_BLT_EXT.3/LE Bluetooth Encryption Parameters (LE)

| | |
|---|---|
| FTP_BLT_EXT.3.1/LE | The TSF shall set the minimum encryption key size to [*128 bits*] for [LE] and not negotiate encryption key sizes smaller than the minimum size. |

## 4.4 Assurance Requirements

23 The TOE security assurance requirements are summarized in Table 17.

**Table 17: Assurance Requirements**

| Assurance Class | Components | Description |
|---|---|---|
| ASE: Security Target | ASE_CCL.1 | Conformance Claims |
| | ASE_ECD.1 | Extended Components Definition |
| | ASE_INT.1 | ST Introduction |
| | ASE_OBJ.2 | Security Objectives |
| | ASE_REQ.2 | Derived Security Requirements |
| | ASE_SPD.1 | Security Problem Definition |
| | ASE_TSS.1 | TOE Summary Specification |
| ADV: Development | ADV_FSP.1 | Basic Functional Specification |
| AGD: Guidance Documentation | AGD_OPE.1 | Operational User Guidance |
| | AGD_PRE.1 | Preparative Procedures |
| ALC: Life-Cycle Support | ALC_CMC.1 | Labelling of the TOE |
| | ALC_CMS.1 | TOE CM Coverage |
| | ALC_TSU_EXT.1 | Timely Security Updates |
| ATE: Tests | ATE_IND.1 | Independent Testing – Conformance |
| AVA: Vulnerability Assessment | AVA_VAN.1 | Vulnerability Survey |

# 5 TOE Summary Specification

## 5.1 Security Audit – (FAU)

### 5.1.1 FAU_GEN.1 Audit Data Generation

24 Audit events are generated for the following audit functions:

a) Start-up and shut-down of the audit functions

b) Authentication events (Success/Failure)

c) Use of privileged/special rights events (Successful and unsuccessful security, audit, and configuration changes)

d) Privilege or role escalation events (Success/Failure)

e) Administrator or root-level access events (Success/Failure)

25 Each audit record contains the following information:

a) Date and time

b) Type of event

c) Subject identity (if applicable)

d) Outcome (success or failure)

26 The logging system captures messages across all levels of the system using two logging components: auditd and Apple Unified Logging. The TOE stores the log data in memory and data store on disk. Audit events are only accessible by administrators. Audit events can be viewed via the command line using their respective tools described in the Administrator Guidance. Within auditd, if the username is longer than 8 characters, the system uses the user ID to display. To access the audit logs generated using auditd, the TOE provides built-in utilities "audit", "praudit", and "auditreduce". All audit records are Basic Security Module (BSM) compliant, and any BSM Audit Tool could be used for viewing audit logs. To access the audit logs generated using the Apple Unified Logging framework, the TOE provides the built-in utility “log”.

### 5.1.2 FAU_GEN.1/BT Audit Data Generation (Bluetooth)

27 The TOE generates Bluetooth audit records after the "Bluetooth for macOS" configuration profile is installed on the TOE. This profile is obtainable from the Apple Developer website under "Profiles and Logs » macOS" along with instructions. The administrator can use Apple Unified Logging to retrieve the relevant events.

28 The TOE audits the following events:

a) Start-up and shutdown of the audit functions

b) Specifically defined auditable events listed in Table 11.

29 Each audit record contains the following information:

a) Date and time

b) Type of event

c) Subject identity

d) Outcome (success or failure)

e) Additional information as specified in Table 11.

30 The TOE does not generate audit records for Bluetooth duplicate connection attempts (FIA_BLT_EXT.3) because rejections happen at the Host Controller Interface (HCI) layer.

### 5.1.3 FAU_GEN.1/WLAN Audit Data Generation (Wireless LAN)

31 The TOE implements functionality to generate audit records for the following auditable events:

a) Startup and shutdown of the audit functions;

b) All auditable events for [*not specified*] level of audit; and

c) Auditable events specified in Table 12 and selected SFRs in Table 13.

32 Each audit record contains the following information:

a) Date and time

b) Type of event

c) Subject identity

d) Outcome (success or failure)

e) Additional information as specified in Table 12 and Table 13.

33 The administrator can use Apple Unified Logging to retrieve the relevant events. An example audit record looks like this:

```
timestamp thread type activity pid ttl message
```

34 The `timestamp` represents when the event occurred. The `thread` is used to represent the specific thread that generated the log message. The `type` represents (among other things) the log severity indicator. The `activity` is an ID that links entries across process boundaries, if needed. The `pid` identifies the specific process that generated the entry. The `ttl` indicates whether a log message has a specific retention time. The `message` component contains specific information of the event in question related to processes, subsystems, and libraries, as well as containing a human-readable description of the event.

## 5.2 Cryptographic Support - (FCS)

### 5.2.1 FCS_CKM.1 Cryptographic Key Generation, FCS_CKM.1/WPA Cryptographic Key Generation (Symmetric Keys for WPA2/WPA3 Connections)

35 The TOE supports generation of 3072-bit, and 4096-bit RSA keys conforming to FIPS PUB 186-4 "Digital Signature Standard (DSS)", Appendix B.3. The TOE provides RSA key generation for being used in TLS sessions with client authentication.

36 The TOE supports NIST curves P-256, P-384, and P-521 for key generation conforming to FIPS PUB 186-4 "Digital Signature Standard (DSS)", Appendix B.4. The TOE provides ECDSA key generation for use in TLS sessions with client authentication and generates ephemeral keys using P-384 and P-521 curves during ECDH key establishment. Bluetooth SSP uses ephemeral ECDH curve P-256 for key establishment.

37 Please refer to Table 18 for details.

38 The TOE generates symmetric keys for Wi-Fi connections in accordance with PRF-384 and PRF-704 as defined in IEEE 802.11-2012 and updated by IEEE 802.11ac-2013 to accommodate AES-GCMP-256. It is implemented in the TOE as part of the WPA implementation and used for the generation of AES keys of 256-bits. These keys are held in RAM only.

39 Apple performs in-house testing of the cryptographic implementation as well as obtaining relevant CAVP certificates necessary to meet requirements.

### 5.2.2 FCS_CKM.2 Cryptographic Key Establishment

40 The TOE supports cryptographic key establishment using the following schemes:

a) RSAES-PKCS1-v1_5 RSA-based key establishment with 3072-bit, and 4096-bit keys as specified in Section 7.2 of RFC 8017.

b) Elliptic curve-based key establishment with NIST curves P-256, P-384, and P-521 as specified in NIST SP 800-56A rev 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".

41 RSA-based key establishment is used in TLS sessions when ciphersuites with RSA key exchange are negotiated. The TOE acts as the sender for RSA-based key establishment schemes. When the TOE detects a decryption error, it warns the invoker that an error occurred, but it does not provide any cryptographically-sensitive data to the invoker or in log files to unauthorized users.

42 Elliptic curve-based key establishment (P-384, and P-521) is used in TLS sessions when ciphersuites with ECDHE key exchange are negotiated. Bluetooth SSP initialization also uses elliptic curve-based key establishment (P-256).

43 Please refer to Table 18 for details.

### 5.2.3 FCS_CKM.2/WLAN Cryptographic Key Distribution (Group Temporal Key for WLAN)

44 The TOE performs Group Temporal Key (GTK) decrypt in accordance with the following key distribution method: AES Key Wrap (as defined in RFC 3394) in an EAPOL-Key frame (as defined in IEEE 802.11-2012) and does not expose the cryptographic keys.

45 AES key wrapping used for GTK is sent in an EAPOL key frame in message three of the 4-way handshake defined in section 11.6.2 of IEEE 802.11-2012.

### 5.2.4 FCS_CKM_EXT.4 Cryptographic Key Destruction

46 On the TOE, each file is protected with a unique per-file key. The key, wrapped using AES-KW, is further wrapped with one of several class keys, depending on how the file is meant to be accessed.[2] The wrapped per-file key is then stored in the file's metadata.

47 When a file is requested to be opened by the kernel API, its metadata is decrypted by the file system key, revealing the wrapped per-file key. The per-file key is unwrapped with a file protection class key and used to decrypt the file as it's read from non-volatile storage. All wrapped file key handling occurs in the SEP; the file key is never directly exposed to the Application Processor.

48 The metadata of all files in the data volume file system is encrypted with an "effaceable key" (a KEK), which is created when the operating system is first installed or when the device is wiped by a user. This key is used to wrap the file system encryption key (also a KEK). The effaceable key doesn't provide additional confidentiality of data; rather, it's designed to be quickly erased on demand. Erasing the effaceable key renders all files cryptographically inaccessible. The effaceable key never leaves the SEP.

49 Within the TOE application space, there are several long-term private keys of note available to be managed by end-users and the system:

a) General-purpose TLS private keys;

b) Bluetooth keys; and

c) WLAN client EAP-TLS private keys.

50 The TOE includes a Keychain Access program that allows users the ability to add, remove, and manage certificates, shared secrets, and the private keys mentioned above. The keys are stored within a Keychain database file, which is protected by a file system DEK on the volume, and further protected within the Keychain database by another set of user-space KEKs and DEKs.

[2] A "class key" is one of three protection classes (Classes A, B, and C) meant to enforce file protection policies to determine when data on the volume is accessible. The class keys are stored and managed by the SEP and are never exposed to the Application Processor (i.e. the kernel or user-space).

51 Please see Section 5.2.10 below for Keychain details. Each Keychain item is protected with an individual DEK. The metadata of all Keychain items is collectively encrypted with another DEK. The KEKs are derived through the process described in section 5.4.2, The encryption key on the smart card (described in section 5.4.2) and all DEKs are generated using approved DRBG(s) per section 5.2.9.

52 Each DEK is wrapped by the Secure Enclave using a class key acting as the key encryption key (KEK) as requested by the creator of the Keychain item. The KEK remains inside the Secure Enclave. The KEKs are wrapped by a key derived from the user's password.

53 TLS private keys and certificates are stored encrypted in Keychain file. TLS session keys are generated during the TLS handshake process and are introduced into volatile memory for the duration of the TLS session. These keys are used to encrypt and decrypt data transmitted over the TLS connection. TLS session keys in volatile memory are zeroized when no longer needed.

54 Bluetooth SSP private keys and data encryption keys are not stored persistently. Rather, the Bluetooth "Link Key" is stored encrypted in the keychain. The Link Key is used to ensure persistent bonding and authentication to Bluetooth devices. Additionally, the Link Key is used to derive the Bluetooth data encryption key.

55 All Keychain DEKs and KEKs are always stored in wrapped (encrypted) form in non-volatile storage. The unwrapped copy of the key is solely held in volatile memory for the duration that key is required to unwrap the DEK (for KEKs) or to decrypt the data (for DEKs).

56 The TOE destroys keys in non-volatile memory by instructing the SEP to erase the effaceable key. Erasing the effaceable key renders all files cryptographically inaccessible. Even though some application-level keys, such as those used by Keychain Access, are partly derived from the user's password, the Keychain database is stored on the encrypted data volume. Once the effaceable key is destroyed, the data volume becomes cryptographically inaccessible and there is no way to access the Keychain database, even if the user password is known.

57 Once the cryptographic operations are complete (at the end of the TLS session, the end of the Bluetooth connection, or the end of a WLAN connection) ephemeral keys are securely destroyed by overwriting them with zeros in volatile memory.

58 The TOE also provides an interface to add, remove and manage certificates and private keys through the Keychain Services API, which is described in [CCGUIDE].

**Table 18: Cryptographic algorithm table**

| SFR | Algorithm | Capabilities | Usage |
|---|---|---|---|
| FCS_CKM.1 | RSA KeyGen | 3072, 4096 | Client authentication in TLS/EAP-TLS client with mutual authentication |

| SFR | Algorithm | Capabilities | Usage |
|---|---|---|---|
| | ECDSA KeyGen | P-256, P-384, P-521 | Bluetooth SSP (P-256)<br>Client authentication in TLS/EAP-TLS client with mutual authentication and key establishment using ephemeral keys in TLS client (ECDHE) (P-384, P-521) |
| FCS_CKM.2 | RSA RSAES-PKCS1-v1_5 Key Establishment | 3072, 4096 | Key establishment in TLS/EAP-TLS client |
| | ECC Key Establishment (KAS-ECC) | P-256, P-384, P-521 | Bluetooth SSP (P-256)<br>Key establishment in TLS/EAP-TLS client (P-384, P-521) |
| FCS_COP.1 /ENCRYPT | AES-KW | 256-bit | Secret key Encryption<br>AES Key Wrap for GTK |
| | AES-CTR | 256-bit | Supporting wireless traffic encryption |
| | AES-GCM | 256-bit | Keychain protection<br>TLS/EAP-TLS client |
| | AES-CCM | 128-bit | Bluetooth SSP (AES-CCM-128) |
| | AES-GCMP-256 | 256-bit | Encrypting wireless traffic |
| FCS_COP.1/HASH | SHA-256,<br>SHA-384, SHA-512 | | Secure boot SHA-512<br>TLS/EAP-TLS client<br>Trusted update SHA-512 |
| FCS_COP.1/SIGN | RSA SigGen/SigVer | 3072, 4096<br>with:<br>SHA-256,<br>SHA-384, SHA-512 | TLS/EAP-TLS client |

| SFR | Algorithm | Capabilities | Usage |
|---|---|---|---|
| | ECDSA SigGen/SigVer | P-384, P-521<br>with:<br>SHA-256,<br>SHA-384, SHA-512 | Secure boot (P-521 with SHA-512)<br>TLS/EAP-TLS client<br>Trusted update (P-521 with SHA-512) |
| FCS_COP.1/ KEYHMAC | HMAC-SHA-256,<br>HMAC-SHA-384 | | Bluetooth BR/EDR (HMAC-SHA-256)<br>Self-Tests (HMAC-SHA-256)<br>TLS/EAP-TLS client (HMAC-SHA-384) |
| FCS_RBG_EXT.1 | CTR_DRBG | AES | Bluetooth<br>TLS/EAP-TLS client |

### 5.2.5 FCS_COP.1/ENCRYPT Cryptographic operation - encryption/decryption

59 The TOE supports AES encryption using 128-bit and 256-bit keys in the following modes:

a) CTR (256-bit) as specified in NIST SP 800-38A (supporting Wi-Fi encryption)

b) GCM (256-bit) as specified in NIST SP 800-38D (for TLS/EAP-TLS, Keychain)

c) AES-GCMP-256 (as defined in NIST SP 800-38D and IEEE 802.11ac-2013) (for Wi-Fi encryption)

d) CCM (128-bit) as specified in NIST SP 800-38C (for Bluetooth only)

e) KW (256-bit) as specified in NIST SP 800-38F (for secret key encryption and GTK Key Wrap)

60 Please refer to Table 18 for details.

### 5.2.6 FCS_COP.1/HASH Cryptographic operation – hashing

61 The TOE supports cryptographic hashing services conforming to FIPS Pub 180-4. The hashing algorithms are used for signature services and HMAC services.

62 The TOE supports the following hash algorithms: SHA-256, SHA-384, and SHA-512. The message digest sizes supported are: 256 bits, 384 bits, and 512 bits.

63 Please refer to Table 18 for details.

### 5.2.7 FCS_COP.1/SIGN Cryptographic operation – signing

64 The TOE provides cryptographic signature generation and verification in accordance with the following cryptographic algorithms:

a) RSA digital signature algorithm conforming to FIPS Pub 186-4, "Digital Signature Standard (DSS)", Section 4. The RSA key sizes supported are: 3072 bits, and 4096-bit.

b) Elliptical curve digital signature algorithm conforming to NIST SP 800-186, "Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters", Section 3. The TOE supports NIST curves P-384, and P-521.

65 Please refer to Table 18 for details.

### 5.2.8 FCS_COP.1/KEYHMAC Cryptographic operation - keyed-hash message authentication

66 The TOE supports keyed-hash message authentication conforming to FIPS Pub 198-1 "The Keyed-Hash Message Authentication Code" and FIPS Pub 180-4 "Secure Hash Standard" with the following algorithms:

a) HMAC-SHA-256

b) HMAC-SHA-384

67 The TOE supports key sizes 256, and 384 bits for all HMAC algorithms.

68 Please refer to Table 18 for details.

### 5.2.9 FCS_RBG_EXT.1 Random Bit Generation

69 The TOE uses a CTR_DRBG(AES) to generate random bits. The source of entropy in the Secure Enclave of the TOE is the Apple SEP TRNG entropy source [ESV E113], which seeds an SP800-90A-compliant Counter DRBG with CAVP #A3490. The DRBG generates 256-bit random numbers for the purpose of key generation in the Secure Enclave. The source of entropy for the main processor of the TOE is the Corecrypto Non-Physical Entropy Source [ESV E181]. The Corecrypto Non-Physical Entropy Source seeds the Corecrypto DRBG with 512 bits of entropy. The TOE's Corecrypto DRBG is SP800-90A-compliant Counter DRBG and it has CAVP #A5946 (Apple Silicon). The operating system uses the XNU hybrid kernel for delivering the entropy from the entropy source to seed and reseed the DRBG. Both the kernel and the user space modules are derived from the same Corecrypto source code.

70 Please refer to Table 18 for details.

### 5.2.10 FCS_STO_EXT.1 Storage of sensitive data

71 The TOE stores the following sensitive data:

a) Trusted Certificates are used for establishing TLS sessions and are stored in the macOS Keychain.

b) Private Keys are used for establishing TLS session and are stored in the macOS Keychain.

c) Bluetooth Link Keys are used to manage the pairing and secure communication with Bluetooth devices and are stored in the macOS Keychain.

72 The TOE does not store login passwords. Instead, the TOE converts the user's typed password into a key used to unwrap the user's class keys. This process is described in section 5.4.2.

73 Each class key is managed by the Secure Enclave and is unwrapped by the key that is produced by PBKDF2 of the user password. If the typed password is correct (ie. It successfully unwraps the user's class keys), the user is considered authenticated; otherwise, authentication fails.

74 TOE usernames are handled via the "Users & Group" GUI described in section 4.2 of the [CCGUIDE].

75 The TOE offers a repository, called Keychain, that provides apps a convenient and secure location to store trusted certificates and private keys. It can be accessed by opening the Keychain Access app in the /System/Applications/Utilities/ folder. An initial default keychain is created for each user, though users can create other keychains for specific purposes.

76 In addition to user keychains, the TOE relies on a number of system-level keychains that maintain authentication assets that are not user-specific, such as network credentials and public key infrastructure (PKI) identities. The Keychain items are encrypted using two different AES-256-GCM keys: a table key (metadata), and a per-row key (secret-key). Keychain metadata (all attributes other than kSecValue) is encrypted with the metadata key to speed searches while the secret value (kSecValueData) is encrypted with the secret-key. The metadata key is protected by the Secure Enclave but is cached in the Application Processor volatile memory to allow fast queries of the keychain. The secret key always requires a derived KEK to unwrap it. Unwrapped KEKs are stored in the protected memory of the Secure Enclave.

77 When the user locks the screen, logs out, or the device automatically locks, the KEK wrapping key and data keys are discarded (i.e., zeroized), rendering all data inaccessible until the user enters the password again.

78 Applications can use the Keychain Services API described in [CCGUIDE] to manage trusted certificates and private keys.

### 5.2.11 FCS_CKM_EXT.8 Bluetooth key generation

79 The TOE generates a new ECDH key pair for every new Bluetooth connection attempt. Static ECDH key pairs are not permitted.

### 5.2.12 FCS_TLS_EXT.1 TLS Protocol, FCS_TLSC_EXT.1 TLS Client Protocol, FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication, FCS_TLSC_EXT.5 TLS Client Support for Supported Groups Extension

80 The TOE implements TLS version 1.2 and rejects all earlier TLS versions. The TOE supports the following ciphersuites:

- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

81 The TOE establishes the reference identifier by parsing the DNS Name for the configured TLS server. The reference identifier is matched against the SAN. The TOE requires the SAN to be presented in the server's certificate for proper validation. The TOE supports wildcards in the DNS name of the server certificate. The TOE does not support URI reference identifiers, or SRV reference identifiers. The TOE supports certificate pinning for trusted update servers only. These pinned trust anchors cannot be adjusted by administrators to prevent use of anything other than official Apple update servers being used. The TOE does not support IP addresses as reference identifiers in the evaluated configuration.

82 The TOE will not establish a TLS connection if the server certificate is invalid.

83 The TOE supports mutual authentication using X.509v3 certificates and, when configured, will send a Certificate and Certificate Verify message in response to a Certificate Request message from a TLS server.

84 The TOE supports the Supported Groups Extension in Client Hello messages by default with the following supported groups: secp384r1, secp521r1.

### 5.2.13 FCS_TLSC_EXT.1/WLAN TLS Client Protocol (EAP-TLS for WLAN), FCS_TLSC_EXT.2/WLAN TLS Client Support for Supported Groups Extension (EAP-TLS for WLAN)

85 For the Wi-fi EAP-TLS client, the TOE implements TLS version 1.2 (RFC 5246) and rejects all earlier TLS versions in support of the EAP-TLS protocol as specified in RFC 5216. All random values used in the EAP-TLS exchange use the RBG specified in FCS_RBG_EXT.1.

86 The TOE supports the following ciphersuites:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289.

87 The TOE establishes the reference identifier by parsing the DNS Name for the configured TLS server. The reference identifier is matched against the CN or the SAN. The TOE does not support wildcards in the DNS name of the server certificate. The TOE does not support URI reference identifiers, or SRV reference identifiers. The TOE does not support IP addresses as reference identifiers in the evaluated configuration.

88 The TOE will not establish a TLS connection if the server certificate is invalid. The TOE also provides the ability for an administrator to configure the list of CA's that are allowed to sign authentication server certificates that are accepted by the TOE.

89 The TOE Wi-Fi client for 802.1X EAP-TLS requires mutual authentication using X.509v3 certificates and will send a Certificate and Certificate Verify message in response to a Certificate Request message from a TLS server.

90 The TOE supports the Supported Groups Extension in Client Hello messages by default with the following supported groups: secp384r1.

### 5.2.14 FCS_WPA_EXT.1 Supported WPA Versions

91 The TOE supports the use of the Wi-Fi Protected Access 3 (WPA3) security type when connecting to wireless networks.

## 5.3 User Data Protection - (FDP)

### 5.3.1 FDP_ACF_EXT.1 Access controls for protecting user data

92 The TOE uses the Apple File System (APFS), which provides access control to data. File system object attributes includes manipulation of metadata (e.g., change, access, modify time) as well as owner and permission data (e.g., group IDs for allowing multiple users to have the same access privileges, user IDs for individual access privileges, and permissions that can be assigned per user or group). The TOE provides the following file system security schemes: sandbox entitlements, POSIX access control lists (ACLs), Unix (BSD) permissions, and per-file BSD flags that override Unix permissions.

93 These security schemes fit together as follows (rules are processed in the following order):

a) If the app's sandbox forbids the requested access, the request is denied.

b) If ownership checking has been disabled for the volume in question by the system administrator (with a checkbox in its Finder Get Info window), the request is granted.

c) If an ACL exists on the file, it is evaluated and used to determine access rights.

d) If a file flag prohibits the operation, the operation is denied.

e) Otherwise, if the user ID matches the owner of the file, the Unix "user" permissions (also called "owner" permissions) are used.

f) Otherwise, if the group ID matches the group for the file, the Unix "group" permissions are used.

g) Otherwise, the Unix "other" permissions are used.

#### Sandbox Entitlements

94 The TOE supports the use of a sandbox to limit an app's ability to access files. These limits override any permissions the app might otherwise have. Sandbox limits are subtractive, not additive. Therefore, the file system permissions represent the maximum access an app might be allowed if its sandbox also permits that access.

#### POSIX ACLs

95 The TOE by default does not restrict access to files created in a user's home directory to the owner only. To enforce this restriction, as part of the [CCGUIDE], the administrator is required to adjust the permissions of newly created user home directories as well as institute a persistent permissions mask (called the "umask").

96 Setting the umask will ensure that any new files or directories, by default, will have permissions set to 700 for directories and 600 for files, meaning that only the owner of the resource has read, write, and execute permissions.

97 The TOE supports ACLs, which are data structures that provide much more detailed control over permissions than Unix permissions. For example, ACLs allow the system administrator to specify that a specific user can delete a file but cannot write to it. An ACL consists of an ordered list of access control entries (ACE), each of which associates a user or group with a set of permissions and specifies whether each permission is allowed or denied. ACEs also include attributes related to inheritance.

98 Each ACE in a directory's ACL can contain any combination of the following inheritance flags:

a) Inherited (this ACE was inherited)

b) File Inherit (this ACE should be inherited by files created within this directory)

c) Directory Inherit (this ACE should be inherited by directories created within this directory)

d) Inherit Only (this ACE should not be checked during authorization)

e) No Propagate Inherit (this ACE should be inherited only by direct children; that is, the ACE should lose any Directory Inherit or File Inherit bit when inherited)

99 When it creates a new file, the kernel goes through the entire access control list of the parent directory and copies to the file's ACL any ACEs that are marked for file inheritance. Similarly, when it creates a new subdirectory, the kernel copies to the subdirectory's ACL any ACEs that are marked for directory inheritance.

100 If a file is copied and pasted into a directory, the kernel replicates the contents of the source file into a new file at the destination. Because it is creating a new file, the system checks the ACL of the parent directory and adds any inherited ACEs to whatever ACEs were in the original file. If a file is moved into a directory, on the other hand, the original file is not replicated and no ACEs are inherited. In this case, the parent directory's ACEs are added to the moved file only if the administrator specifically propagates ACEs from the parent directory through contained files and subdirectories. Similarly, once a file has been created, changing the ACL of the parent directory does not affect the ACL of contained files and subdirectories unless the administrator specifically propagates the change.

101 In BSD, applying a directory's permissions to enclosed files and subdirectories completely replaces the permissions of the enclosed objects. With ACLs, in contrast, inherited ACEs are added to other ACEs already on the file or directory.

102 The order in which ACEs are placed in an ACL and therefore the order in which they are evaluated to determine permissions is as follows:

a) Explicitly specified deny associations

b) Explicitly specified allow associations

103 Inherited associations appear in the same order in which they appeared in the parent. Since ACEs can be inherited, administrators can control the fine-grained permissions of files created in a directory by assigning inheritable ACEs to the directory.

#### Unix Permissions

104 Each file system object has a set of UNIX permissions defined by three attributes

a) UID, short for User ID. Commonly referred to as the File's Owner.

b) GID, short for Group ID.

c) Flags that include permission bits and other related attributes.

105 The flags for a file or directory are a 16-bit value that is often represented as a three-digit or four-digit octal value (with the most significant bits set to zero when not defined). The Owner, Group, and Other bit sets contain three bits: read, write, execute (rwx for short). In addition, there are setuid, setgid, and sticky bits which can affect permissions on files and directories in certain conditions.

#### BSD File Flags

106 In addition to the standard Unix file permissions, the TOE supports several BSD file flags provided by the chflags API and the related chflags command. These flags override the Unix permissions.

107 The TOE also allows admin users to disable ownership and permissions checking for removable volumes on a per-volume basis by choosing Get Info on the volume in Finder then checking the "Ignore ownership on this volume" checkbox.

## 5.4 Identification and Authentication - (FIA)

### 5.4.1 FIA_AFL.1 Authentication failure handling

108 The TOE will detect when an administrator-configurable integer within 1-50 unsuccessful authentication attempts have been met when using username and password, or username and PIN mechanisms for authentication. Once the specified number of unsuccessful authentication attempts for an account has been met, the TOE will lock out the account for an administrator configured period of time.

### 5.4.2 FIA_UAU.5 Multiple authentication mechanisms

109 The TOE supports authentication based on username/password and username/smart card.

110 For password-based authentication, the user account requires a username and a password credential. To initiate the authentication process, the user enters their username and is prompted for a password. A 256-bit key is derived from the user password using the Password-Based Key Derivation Function 2 (PBKDF2) with SHA-256. This key is then used to unwrap (decrypt) the user class key (KEK) using AES-KW.

If the output key from the PBKDF2 function successfully unwraps the user's class key, the user is authenticated and granted access; otherwise, the user is denied access.

111 For smart card authentication, the user's smart card must first be registered with the TOE and be associated with the user. Upon registration, the smart card is provisioned with a digital certificate and an encryption key (a key-encrypting-key in PIV smart cards slot 9d called the "key management key"). This encryption key is generated using the approved DRBGs described in FCS_RBG_EXT.1. When the user inserts the smart card to authenticate, the user enters the associated PIN to unlock and access the certificate and encryption key. Once unlocked, a signing operation is performed by the card and the TOE verifies the signature using the paired certificate for authentication. The key management key is then used to unwrap the keychain.

### 5.4.3 FIA_X509_EXT.1 X.509 Certificate Validation, FIA_X509_EXT.1/WLAN X.509 Certificate Validation

112 When an X.509 certificate is presented for authentication, the TOE verifies the certificate path and certification validation process by verifying the following rules:

a) RFC 5280 certificate validation and certificate path validation.

b) The certificate path must terminate with a trusted CA certificate.

c) The OS shall validate a certificate path by ensuring the presence of the basicConstraints extension, that the CA flag is set to TRUE for all CA certificates, and that any path constraints are met.

d) The TSF shall validate that any CA certificate includes caSigning purpose in the key usage field

e) The OS shall validate the revocation status of the certificate using OCSP (except for 802.1X EAP authentication). The certificate is accepted if its revocation status cannot be determined.

113 For 802.1X EAP authentication, no revocation checking is performed.

114 The TOE validates the extendedKeyUsage field depending on the specific usage of the certificate as follows:

a) Certificates used for trusted updates and executable code integrity verification shall have the Code Signing Purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.

b) Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.

c) Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the EKU field.

d) S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the EKU field.

e) OCSP certificates presented for OCSP responses shall have the OCSP Signing Purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the EKU field.

f) Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kpcmcRA with OID 1.3.6.1.5.5.7.3.28) in the EKU field.

115 X.509 certificates are validated when imported into the TOE's trusted certificate store, during session establishment with a peer, and prior to presenting a certificate to the peer during trusted channel implementation using TLS for mutual authentications.

116 As described, the extendedKeyUsage (EKU) is validated against the requirements defined in FIA_X509_EXT.1 (which is a superset of the requirements of FIA_X509_EXT.1/WLAN).

### 5.4.4 FIA_X509_EXT.2 X.509 Certificate Authentication, FIA_X509_EXT.2/WLAN X.509 Certificate Authentication (EAP-TLS for WLAN)

117 The TOE is capable of using X.509v3 certificates for performing mutual authentication for TLS connections.

118 The TOE also uses X.509v3 certificates as defined by RFC 5280 to support authentication for 802.1X EAP-TLS exchanges.

### 5.4.5 FIA_X509_EXT.6 X.509 Certificate Storage and Management

119 The TOE ensures all certificates on the system are securely stored and protected from unauthorized deletion or modification. The database containing trust anchors for all certificates is protected via integrity check and write protection. Only authorized administrators can load trust anchors into the system trust store.

### 5.4.6 FIA_PAE_EXT.1 Port Access Entity Authentication

120 The TOE conforms to IEEE Standard 802.1X for a Port Access Entity (PAE) in the “Supplicant” role.

### 5.4.7 FIA_BLT_EXT.1 Bluetooth User Authorization

121 The TOE supports SSP and the following Bluetooth association models:

a) Numeric comparison

b) Passkey entry

122 Users can pair their TOE device with a remote Bluetooth device using the 'Set up Bluetooth Device' option from the Bluetooth status menu. Users can also remove a device from the TOE's device list. Explicit user authorization is required for both pairing and removing a Bluetooth device from the TOE's device list.

123 Manual authorization is implicitly configured since pairing can only occur when explicitly authorized through the System Settings » Bluetooth interface of the TOE device. During the pairing time, another device (or the TOE) can send a pairing request. Commonly, a six-digit number is displayed on both sides, which must be manually matched by a user (i.e., the PIN is shown and the user must accept it before the pairing completes). If one device does not support this automatic exchange of a

PIN, a window for entering a manual PIN is shown. That PIN must match on both sides. This also applies to applications that use Bluetooth.

124 Bluetooth devices such as Keyboards, mice, trackpads, etc. can also be paired with the TOE simply by connecting the device with a USB cable while the user is successfully logged in.

### 5.4.8 FIA_BLT_EXT.2 Bluetooth Mutual Authentication

125 The TOE's Bluetooth device driver prevents data transfer via Bluetooth until pairing has fully completed and the devices have mutually authenticated.

126 The TOE supports the Logical Link Control and Adaptation Layer Protocol (L2CAP) through an API in the IOBluetoothDevice class.

127 An RFCOMM channel object can be obtained by opening an RFCOMM channel in a device or by requesting a notification when a channel is created (this is commonly used to provide services). See the IOBluetooth RFCOMMChannel class.

### 5.4.9 FIA_BLT_EXT.3 Rejection of Duplicate Bluetooth Connections

128 Bluetooth devices may not establish more than one connection. Multiple connection attempts (i.e., pairing and session initialization attempts) from the same BD_ADDR for an established connection will be discarded. For details of the security of Bluetooth/LE see the Bluetooth Specifications at https://www.bluetooth.com/specifications/.

129 Rejection is performed at the Host Controller Interface (HCI) layer.

### 5.4.10 FIA_BLT_EXT.4 Secure Simple Pairing

130 Devices that want to pair with the TOE via Bluetooth can use Secure Simple Pairing. The TOE implements SSP which is the 2nd generation of the security key exchange scheme to use ECDH-based key exchange using NIST curve P-256. Message integrity and data protection is supported with the use of AES and 128-bit keys.

131 The SSP process uses elliptic curve Diffie-Hellman algorithms to establish a shared secret between the TOE and Bluetooth devices supporting authentication and encryption of exchanged data.

### 5.4.11 FIA_BLT_EXT.6 Trusted Bluetooth device user authorization

132 The TOE supports Bluetooth including Basic Rate/Enhanced Data Rate (BR/EDR) and Low Energy (LE) with the following Bluetooth profiles:

a) Hands-Free Profile (HFP 1.6)

b) Phone Book Access Profile (PBAP)

c) Advanced Audio Distribution Profile (A2DP)

d) Audio/Video Remote Control Profile (AVRCP 1.4)

e) Personal Area Network Profile (PAN)

f) Human Interface Device Profile (HID)

g) Message Access Profile (MAP)

133 Users can pair their TOE device with a remote Bluetooth device using the 'Set up Bluetooth Device' option from the Bluetooth status menu. They can also remove a device from the TOE's device list. Explicit user authorization is required for both pairing and removing a Bluetooth device from the TOE's device list.

134 Manual authorization is implicitly configured since pairing can only occur when explicitly authorized through the System Settings » Bluetooth interface. During the pairing time, another device (or the TOE) can send a pairing request. Commonly, a six-digit number is displayed on both sides, which must be manually matched by a user (i.e., the PIN is shown and the user must accept it before the pairing completes). If one device does not support this automatic exchange of a PIN, a window for entering a manual PIN is shown. That PIN must match on both sides.

135 The TOE automatically authorizes the remote Bluetooth device during pairing for all Bluetooth profiles the remote device announces to support during the pairing operation. This approach avoids user confusion between a paired device to which the TOE is connected and authorized and thus can communicate with and a device to which the TOE is connected but not yet authorized with which the TOE cannot yet communicate. To de-authorize a device, the user would unpair the device. The TOE establishes a "trusted relationship" with an authorized device at the time of pairing. The only difference in behavior between a trusted device and an untrusted device is that the untrusted device must first be manually authorized as described in the previous paragraph.

### 5.4.12 FIA_BLT_EXT.7 Untrusted Bluetooth device user authorization

136 See section 5.4.11.

## 5.5 Security Management - (FMT)

### 5.5.1 FMT_MOF_EXT.1 Management of security functions behavior

137 See section 5.5.2.

### 5.5.2 FMT_SMF_EXT.1 Specification of management functions

138 The TOE supports the following roles: Administrator and User. The Administrator is a member of the local admin group or an applied configuration profile, and the User is an unprivileged account. Functions requiring Administrator access require the user to enter the correct Administrator password before allowing the user to modify the function.

139 For functions that can be performed by both the Administrator and User roles, if the Administrator overrides the User ability to change the function, the TOE enforces these restrictions by requiring the User to enter the correct Administrator password before allowing the user to modify the function.

**Table 19: TOE Management Functions**

| # | Management Function | Administrator | User |
|---|---|---|---|
| 1 | Enable/disable screen lock | Yes ("Require password after screen saver begins or display is turned off") | Yes ("Require password after screen saver begins or display is turned off") |
| 2 | Configure screen lock inactivity timeout | Yes | Yes (if not restricted by an Administrator) |
| 3 | import keys/secrets into the secure key storage | Yes | No |
| 4 | Configure local audit storage capacity | Yes | No |
| 5 | Configure minimum password Length | Yes | No |
| 6 | Configure minimum number of special characters in password | Yes | No |
| 11 | Configure host-based firewall | Yes | No |
| 15 | Configure audit rules | Yes | No |
| 16 | Configure name/address of network time server | Yes | No |
| 17 | Enable/disable automatic software update | Yes | No |
| 18 | Configure Wi-Fi interface | Yes | No |
| 19 | Enable/disable Bluetooth interface | Yes | No |

140 Audit storage and audit rules can only be configured for the `auditd` component. Apple Unified Logging (`logd`) is managed entirely by the TOE without additional management necessary.

### 5.5.3 FMT_SMF.1/WLAN Specification of Management Functions (WLAN Client)

141 The TOE supports the following management functions pertaining to wireless LAN capabilities.

**Table 20: TOE Management Functions (WLAN)**

| # | Management Function | Admin | User |
|---|---|---|---|
| WL-1 | configure security policy for each wireless network:<br>• [specify the CA(s) from which the TSF will accept WLAN authentication server certificate(s)],<br>• security type,<br>• authentication protocol,<br>• client credentials to be used for authentication, | X | - |
| WL-2 | specify wireless networks (SSIDs) to which the TSF may connect | X | - |
| WL-3 | enable/disable wireless network bridging capability (for example, bridging a connection between the WLAN and cellular radios to function as a hotspot) authenticated by [passcode] | X | - |
| WL-5 | disable ad hoc wireless client-to-client connection capability | X | - |
| WL-8 | loading X.509 certificates into the TOE | X | - |
| WL-9 | revoke X.509 certificates loaded into the TOE | X | - |

**Application Note:** Status markers indicate 'X' (Supported) and '-' (not supported). Optional functions that are not claimed by the TOE are indicated with a 'not supported' status marker and grey shading.

### 5.5.4 FMT_MOF_EXT.1/BT Management of security functions behavior

142 See section 5.5.5.

### 5.5.5 FMT_SMF_EXT.1/BT Specification of management functions

143 The TOE supports both Bluetooth BR/EDR and LE, can make use of Secure Simple Pairing (SSP) for security, and supports the following Bluetooth profiles:

a) Hands-Free Profile (HFP 1.6)

b) Phone Book Access Profile (PBAP)

c) Advanced Audio Distribution Profile (A2DP)

d) Audio/Video Remote Control Profile (AVRCP 1.4)

e) Personal Area Network Profile (PAN)

f) Human Interface Device Profile (HID)

g) Message Access Profile (MAP)

144 See Table 5 for additional information on supported Bluetooth versions.

145 **BT-1**. The TOE allows administrators to enable and disable the Bluetooth Discoverable (for BR/EDR) and Advertising (for LE) modes.

146 The TOE supports all Bluetooth Security Modes and Levels to ensure compatibility and interoperability with a wide range of Bluetooth devices. By default, the TOE will first attempt to negotiate the use of Security Mode 1 Level 4 to provide the most robust security. However, if the peer device does not support Mode 1 Level 4, the TOE will negotiate the highest mode and level supported by the peer device including Security Mode 1 (any level), Security Mode 2 (any level), Security Mode 3 (any level), and Security Mode 4 (levels 0;1;2).

## 5.6 Protection of the TSF – (FPT)

### 5.6.1 FPT_ACF_EXT.1 Access controls

147 The TOE provides access control policies via the System Integrity Protection feature. This technology prevents users, including malicious software and the root user account (an administrator superuser account), from modifying protected files and folders. System Integrity Protection is designed to allow modification of these protected parts only by processes that are signed by Apple and have special entitlements to write to system files, such as Apple software updates and Apple installers.

148 System Integrity Protection protects the following parts of the system from unauthorized modification.

a) Kernel drivers and modules
    i) /System/Library/Extensions/
b) Shared libraries
    i) /Library/Frameworks
    ii) /System/Library/Frameworks
    iii) /System/Library/PrivateFrameworks
c) Applications
    i) /System/Applications Apps that are installed with the TOE and updated as part of TOE updates
    ii) /Applications Apps that are installed with the TOE but updated independently

149 Standard file permissions (FDP_ACF_EXT.1) are used to protect the following from unauthorized modification:

a) Security audit logs
    i) /var/audit
    ii) /var/db/diagnostics

b) System configuration files

i) /Library/Preferences - system-wide "preferences"

ii) ~/Library/Preferences - User-specific "preferences." A user has permission to modify their own configuration files.

iii) /etc/security/audit-control

c) TSF-data

i) /var

ii) /var/db/mds/messages/<PID>/se_SecurityMessages – each se_SecurityMessages file contains messages for the user represented by <PID>. A user has permission to modify their own data.

iii) /var/folders/zz

iv) /var/folders/<non-zz directories> - user-specific TSF data. A user has permission to modify their own data.

d) System-wide credentials repositories (accessed through local directory services)

i) /private/var/db/dslocal/nodes/Default/

e) Applications

i) /Applications Apps installed by a traditional installer and Apps copied into /Applications

150 The TOE prevents unprivileged users from reading Security audit logs and System-wide credential repositories.

### 5.6.2 FPT_ASLR_EXT.1 Address space layout randomization (ASLR)

151 The TOE always randomizes process address memory locations with 16 bits of entropy.

### 5.6.3 FPT_SBOP_EXT.1 Stack buffer overflow protection

152 The TOE protects all TOE binaries from stack-based buffer overflow attacks using:

a) ASLR to randomize locations on the stack, preventing attackers from jumping to specific data that has been written to the stack.

b) Stack canaries to detect if the stack has been overwritten when returning from a function.

153 All TOE binaries are compiled with stack-based overflow protections enabled; however, not all compiled binaries contain stack canaries for one or more of the following reasons:

a) Type 1: The compiler can optimize away stack usage (which macOS heavily relies on for performance reasons).

b) Type 2: Some binaries are just small entry points that rely on system frameworks for all of their functionality. There, the binary itself is going to be

really small (less than ~1000 instructions, sometimes as small as 10 instructions), so is much less likely to need stack protection.

c) Type 3: There are very short program/functions that do not access the stack (and just forward to system frameworks to perform the real work)

d) Type 4: There are tiny binaries (very few instructions) with a single trivial function that do not need stack protections or tiny wrappers that do not make use of the stack.

e) Type 5: Some binaries do not access the stack in any kind of vulnerable way

### 5.6.4 FPT_TST_EXT.1 Boot Integrity, FPT_TST_EXT.3/WLAN TSF Cryptographic Functionality Testing (WLAN Client)

154 The boot process on Apple Silicon devices is as follows:

a) The Application Processor loads the Boot ROM.

b) The Boot ROM validates the Low-Level Bootloader (LLB) signature using the Apple Root CA public key.

c) LLB validates system-paired firmware signatures.

d) LLB validates iBoot stage 2 signature.

e) iBoot stage 2 validates the macOS-paired firmware, Boot Kernel Collection, Auxiliary Kernel Collection (if applicable), system trust cache, and signed system volume signatures.

f) The TOE (macOS) begins execution.

155 The boot process for sepOS is as follows:

a) iBoot assigns a dedicated region of memory to the Secure Enclave.

b) The Application Processor sends the sepOS image to the Secure Enclave Boot ROM.

c) The Secure Enclave Boot ROM checks the digital signature of the sepOS image.

d) If the signature is deemed valid, sepOS begins execution.

156 The Boot ROM for both the Application Processor and the Secure Enclave is immutable code, referred to as the hardware root of trust. It is laid down during chip fabrication and is audited for vulnerabilities and implicitly trusted. The Boot ROM code contains the Apple Root CA public key, which is used to verify the digital signatures of the bootchain.

157 The TOE performs software integrity testing on the runtime image of each implemented module using HMAC-SHA2-256 to calculate a value that is compared with the value stored in the module, computed at compilation time. If the test fails, the module enters an error state where no cryptographic services are provided and data output is prohibited rendering the module non-operational.

158 Integrity tests can be performed on demand by power-cycling the TOE and the test will be performed as part of the per-operational self-tests, and automatically at power-on.

159 The WLAN client relies on cryptographic functionality implemented by the "BC" chip identified in section 7.1 which is embedded in the underlying platform. The TOE verifies that the chip has successfully completed its own integrity self-tests prior to the TSF attempting to use the implementation. While this does present a dependency on the host platform in assessing the assurance provided by these self-tests, the vendor of the TOE OS is also responsible for the host platform thus providing a high level of assurance that the checks are sufficient to ensure the correct functioning of the TSF.

160 Software comprising the WLAN client functionality is cryptographically checked against a static reference hash to ensure it has not been modified. The static reference hash is stored in the secure bootchain, cannot be modified by unauthorized means, and can be relied upon. This mechanism ensures that any unauthorized modification to the stored code will be detected prior to execution, thereby demonstrating that the integrity of the TSF executable code has not been compromised.

### 5.6.5 FPT_TUD_EXT.1 Trusted update

161 The TOE allows the user to check for and install OS updates using the Software Update preference pane. This pane also supports Apple's Rapid Security Responses feature. This feature allows security updates to the TOE to be applied when available without waiting for the next cumulative macOS software update.

162 Signature verification of the TOE image is performed by the SEP prior to the update being installed. If the signature verification fails, the update is not installed, otherwise the update is applied.

163 The TOE includes the Mac App Store app, which allows users to check for and install updates to apps. The TOE validates the digital signature of the apps.

164 On Apple Silicon platforms:

Algorithm: ECDSA P-521 sigver

Standard: NIST SP 800-186 Section 3

Modules:

a) Apple corecrypto Module v14.0 [Apple ARM, User, Software, SL1]

b) Apple corecrypto Module v14.0 [Apple ARM, Kernel, Software, SL1]

c) Apple corecrypto Module v14.0 [Apple Silicon, Secure Key Store, Hardware, SL2]

Signatures are verified using ECDSA P-521 and SHA-512. The CA public key is embedded in the Mac's Boot ROM code during manufacturing. The TOE image is signed using this public key's corresponding private key.

### 5.6.6 FPT_TUD_EXT.2 Trusted update for application software

165 See section 5.6.5.

## 5.7 TOE Access – (FTA)

### 5.7.1 FTA_TAB.1 Default TOE access banners

166 The TOE will display an advisory warning message regarding unauthorized use of the OS prior to establishing a user session.

### 5.7.2 FTA_WSE_EXT.1 Wireless Network Access

167 Administrators can restrict the wireless networks in which the TOE device connects by allowing administrators to define specific wireless networks via SSID the device is allowed to connect to and restricting a user's ability to change those networks.

## 5.8 Trusted Path – (FTP)

### 5.8.1 FTP_ITC_EXT.1 Trusted Channel Communication, FTP_ITC.1/WLAN Trusted Channel Communication (Wireless LAN)

168 The TOE uses TLS as conforming to FCS_TLSC_EXT.1 to provide a trusted channel between itself and authorized IT entities. The update server is the claimed authorized IT entity. (Wireless access points as an authorized IT entity are described separately below.)

169 The TOE ensures trusted communications between itself and a wireless access point by implementing 802.11-2012, 802.1X, and EAP-TLS protocols using FCS_TLSC_EXT.1/WLAN. This communication channel is logically distinct from other communication channels and ensures the identification of its end points and that all channel data is protected against modification and disclosure.

170 The TSF is permitted to initiate communication of the trusted channel, and will always initiate communication of the trusted channel to a wireless access point (WAP).

### 5.8.2 FTP_TRP.1 Trusted Path

171 The TOE provides a trusted path between itself and local users that provides assured identification of its endpoints. The trusted path is restricted to physical access only and is initiated by the local user. Local access is protected by the user's authentication credentials which require physical interaction by an authorized human. The TOE does not support a remote administrative communication path in the evaluated configuration.

### 5.8.3 FTP_BLT_EXT.1 Bluetooth encryption

172 The TOE supports Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) and Low Energy (LE). Bluetooth is enabled by default.

173 Devices that want to pair with the TOE via Bluetooth are required by the TOE to use Secure Simple Pairing, which uses ECDH-based authentication and key exchange and AES for data encryption. This applies to both BR/EDR and LE. Bluetooth encryption is always enabled.

### 5.8.4 FTP_BLT_EXT.2 Persistence of Bluetooth encryption

174 If the remote Bluetooth device stops encrypting while connected to the TOE, the TOE terminates the connection.

### 5.8.5 FTP_BLT_EXT.3/BR Bluetooth encryption parameters (BR/EDR)

175 Connections via BR/EDR and LE are secured using 128-bit AES Counter with CBC-MAC (AES-CCM-128) mode. Although the Bluetooth standard supports the use of 128-bit AES with minimum 8-bit key size to maximum 128-bit key size, The TOE does not support any key sizes smaller than 128-bit; thus, smaller key sizes cannot be negotiated.

176 A shared secret known as a “Link Key” is agreed upon by the two parties using ECDH (NIST P-256). The encryption key is then derived from this shared secret in addition to random numbers (nonces) and addresses exchanged.

### 5.8.6 FTP_BLT_EXT.3/LE Bluetooth encryption parameters (LE)

177 See section 5.8.5 for LE connections.

## 5.9 Timely Security Updates

### 5.9.1 ALC_TSU_EXT.1

178 Apple generally does not disclose, discuss, or confirm security issues until a full investigation is complete and any necessary patches or releases are available. Once an issue has been confirmed and a patch has been made available, references containing technical details on the patches are made available and Common Vulnerabilities and Exposures (CVEs), etc. are released.

179 Apple distributes information about security issues in its products through its "Apple security updates" page (https://support.apple.com/HT201222).

180 Security advisories are also provided through the security-announce mailing list (https://lists.apple.com/mailman/listinfo/security-announce/).

181 Patches are made available via the trusted update server the same day that security issues are announced.

182 Potential security vulnerabilities can be reported by following the procedures on the "Report a security or privacy vulnerability" page (https://support.apple.com/HT201220). This includes sending an email to "productsecurity@apple.com" and includes the ability to encrypt information using the Apple Product Security PGP key (https://support.apple.com/kb/HT201214).

183 The TOE supports Apple's Rapid Security Responses feature. This feature allows security updates to the TOE to be applied when available without waiting for the next cumulative macOS update.

# 6 Rationale

## 6.1 Conformance Claim Rationale

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

a) **TOE type.** As identified in section 2.1, the TOE is a general purpose operating system, consistent with the PP_OS_V4.3. The TOE boundary is simply extended to include Bluetooth functionality per MOD_BT_V1.0 and WLAN Client functionality per MOD_WLANC_V1.0.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the claimed PP, PP Modules, and package in section 1.3.

c) **Security objectives.** As shown in section 3.4, the security objectives are reproduced directly from the claimed PP, PP Modules, and package in section 1.3.

d) **Security requirements.** As shown in section 4, the security requirements are reproduced from the claimed PP, PP Modules, and package in section 1.3. No additional requirements have been specified.

## 6.2 Security Objectives Rationale

185 All security objectives shown in Table 21 are drawn directly from the claimed PP, PP-Modules, and package listed in section 1.3.

**Table 21: Security Objectives Rationale**

| Threat, Assumption, or OSP | Security Objectives | Rationale |
|---|---|---|
| T.NETWORK_ATTACK | O.PROTECTED_COMMS | The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides for integrity of transmitted data. |
| | O.INTEGRITY | The threat T.NETWORK_ATTACK is countered by O.INTEGRITY as this provides for integrity of software that is installed onto the system from the network. |
| | O.MANAGEMENT | The threat T.NETWORK_ATTACK is countered by O.MANAGEMENT as this provides for the ability to configure the OS to defend against network attack. |

| Threat, Assumption, or OSP | Security Objectives | Rationale |
|---|---|---|
| | O.ACCOUNTABILITY | The threat T.NETWORK_ATTACK is countered by O.ACCOUNTABILITY as this provides a mechanism for the OS to report behavior that may indicate a network attack has occurred. |
| T.NETWORK_EAVESDROP | O.PROTECTED_COMMS | The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides for confidentiality of transmitted data. |
| | O.MANAGEMENT | The threat T.NETWORK_EAVESDROP is countered by O.MANAGEMENT as this provides for the ability to configure the OS to protect the confidentiality of its transmitted data. |
| T.LOCAL_ATTACK | O.INTEGRITY | The objective O.INTEGRITY protects against the use of mechanisms that weaken the TOE with regard to attack by other software on the platform. |
| | O.ACCOUNTABILITY | The objective O.ACCOUNTABILITY protects against local attacks by providing a mechanism to report behavior that may indicate a local attack is occurring or has occurred. |
| T.LIMITED_PHYSICAL_ ACCESS | O.PROTECTED_STORAGE | The objective O.PROTECTED_STORAGE protects against unauthorized attempts to access physical storage used by the TOE. |
| T.NETWORK_EAVESDROP (MOD_BT_V1.0) | O.PROTECTED_COMMS | The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides the capability to communicate using Bluetooth as a means to maintain the confidentiality of data that are transmitted outside of the TOE. |

| Threat, Assumption, or OSP | Security Objectives | Rationale |
|---|---|---|
| T.NETWORK_ATTACK (MOD_BT_V1.0) | O.PROTECTED_COMMS | The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides the capability to communicate using Bluetooth as a means to maintain the confidentiality of data that are transmitted outside of the TOE. |
| T.TSF_ FAILURE | O.SELF_ TEST | The threat T.TSF_FAILURE is mitigated by O.SELF_TEST as this defines a mechanism for ensuring the reliability of the TSF by detecting potential failure conditions. |
| T.UNAUTHORIZED_ACCESS | O.AUTH_ COMM | The threat T.UNAUTHORIZED_ACCESS is mitigated in part by O.AUTH_COMM by ensuring the authenticity of any remote endpoint that the TSF connects to. |
| | O.CRYPTOGRAPHIC_ FUNCTIONS | The threat T.UNAUTHORIZED_ACCESS is mitigated in part by O.CRYPTOGRAPHIC_FUNCTIONS by ensuring the confidentiality and integrity of data in transit to protect against man-in-the-middle attacks. |
| | O.TOE_ ADMINISTRATION | The threat T.UNAUTHORIZED_ACCESS is mitigated in part by O.TOE_ADMINISTRATION by using the TOE platform's authentication mechanism to ensure that only authorized administrators can configure the TOE's behavior. |
| | O.WIRELESS_ACCESS_ POINT_CONNECTION | The threat T.UNAUTHORIZED_ACCESS is mitigated in part by this objective because it provides a mechanism to restrict the remote entities that the TOE is permitted to communicate with. |

| Threat, Assumption, or OSP | Security Objectives | Rationale |
|---|---|---|
| T.UNDETECTED_ACTIONS | O.SYSTEM_ MONITORING | The threat T.UNDETECTED_ACTIONS is mitigated by O.SYSTEM_MONITORING by enforcing an auditing mechanism that can be used to track security-relevant TOE behavior. |
| A.PLATFORM | OE.PLATFORM | The operational environment objective OE.PLATFORM is realized through A.PLATFORM. |
| A.PROPER_USER | OE.PROPER_USER | The operational environment objective OE.PROPER_USER is realized through A.PROPER_USER. |
| A.PROPER_ADMIN | OE.PROPER_ADMIN | The operational environment objective OE.PROPER_ADMIN is realized through A.PROPER_ADMIN. |
| A.NO_ TOE_ BYPASS | OE.NO_ TOE_ BYPASS | The operational environment objective OE.NO_TOE_BYPASS is realized through A.NO_TOE_BYPASS. |
| A.TRUSTED_ADMIN | OE.TRUSTED_ADMIN | The Operational Environment objective OE.TRUSTED ADMIN is realized through A.TRUSTED_ADMIN. |

## 6.3 Security Requirements Rationale

186 All security objectives shown in Table 22 are drawn directly from the claimed PP and packages in section 1.3.

**Table 22: SFR Rationale**

| Source | Objective | Addressed by | Rational |
|---|---|---|---|
| PP_OS_V4.3 | O.ACCOUNTABILITY | FAU_GEN.1 | Supports the objective by requiring that critical event information be gathered by the TOE. |
| | | FTP_ITC_EXT.1 | Supports the objective by ensuring that audit information can be securely transmitted to remote systems for analysis. |

| Source | Objective | Addressed by | Rational |
|---|---|---|---|
| MOD_BT_V1.0 | O.ACCOUNTABILITY<br>(GPOS PP only) | FAU_GEN.1/BT | FAU_GEN.1/BT supports the objective by requiring the TSF to specify the Bluetooth-related auditable events for which it will generate audit records. |
| PP_OS_V4.3 | O.INTEGRITY | FPT_SBOP_EXT.1<br>FPT_ASLR_EXT.1 | Supports the objective by requiring that OS applications be hardened against buffer overflow attacks |
| | | FPT_TUD_EXT.1 | Supports the objective by requiring that the OS be able to check for critical updates. |
| | | FPT_TUD_EXT.2 | Supports the objective by requiring that the OS verify updates before applying them. |
| | | FCS_COP.1/HASH | Supports the objective by requiring the TSF to implement hash algorithms that are used in support of protected communications. |
| | | FCS_COP.1/SIGN | Supports the objective by requiring the TSF to implement digital signature algorithms that are used in support of protected communications. |
| | | FCS_COP.1/KEYHMAC | Supports the objective by requiring the TSF to implement HMAC algorithms that are used in support of protected communications. |
| | | FPT_ACF_EXT.1 | Supports the objective by requiring the TSF restrict unprivileged users from changing critical components. |
| | | FIA_X509_EXT.1 | Supports the objective by requiring the TSF to validate |

| Source | Objective | Addressed by | Rational |
|---|---|---|---|
| | | | certificates using industry standards. |
| | | FPT_TST_EXT.1 | Supports the objective by requiring the TSF to verify executable code critical to its operation. |
| | | FTP_ITC_EXT.1 | Supports the objective by requiring the OS to provide a trusted channel for critical communication. |
| | | FIA_AFL.1 | Supports the objective by requiring the TSF to respond accordingly when the number of failed authentication attempts reaches a specified threshold. |
| | | FIA_UAU.5 | Supports the objective by requiring the OS to provide standard authentication mechanisms. |
| MOD_BT_V1.0 | O.INTEGRITY | FAU_GEN.1/BT | FAU_GEN.1/BT supports the objective by requiring the TSF to specify the Bluetooth-related auditable events for which it will generate audit records. |
| PP_OS_V4.3 | O.MANAGEMENT | FMT_MOF_EXT.1 | Supports this objective by requiring the TOE to restrict the ability to perform certain management functions to a privileged user. |
| | | FMT_SMF_EXT.1 | Supports this objective by requiring the TOE to implement specific management functions. |
| | | FTA_TAB.1 | Supports this objective by requiring the TOE to implement a trusted path between the itself and users. |

| Source | Objective | Addressed by | Rational |
|---|---|---|---|
| | | FTP_TRP.1 | Supports this objective by requiring a trusted path between users and the OS. |
| MOD_BT_V1.0 | O.MANAGEMENT<br>(GPOS PP only) | FMT_MOF_EXT.1/BT | FMT_MOF_EXT.1/BT supports the objective by restricting the ability to perform Blue-tooth-related management functions to the Administrator. |
| | | FMT_SMF_EXT.1/BT | FMT_SMF_EXT.1/BT supports the objective by specifying the Bluetooth-related management functions that the TSF must perform. |
| PP_OS_V4.3 | O.PROTECTED_ STORAGE | FCS_STO_EXT.1 | Supports this objective by requiring the OS to provide encrypted storage. |
| | | FCS_RBG_EXT.1 | Supports this objective by requiring the OS to generate random bits according to industry standards. |
| | | FCS_COP.1/ENCRYPT | Supports this objective requiring the OS to perform encryption according to industry stands. |
| | | FDP_ACF_EXT.1 | Supports this objective by requiring the OS to implement access controls. |
| PP_OS_V4.3 | O.PROTECTED_ COMMS | FCS_RBG_EXT.1 | Supports this objective by requiring the OS to generate random bits according to industry standards. |
| | | FCS_CKM.1 | Supports this objective by requiring the TSF to generate asymmetric cryptographic keys to industry standards. |
| | | FCS_CKM.2 | Supports this objective by requiring the TSF to perform key |

| Source | Objective | Addressed by | Rational |
|---|---|---|---|
| | | | establishment according to industry standards. |
| | | FCS_CKM_EXT.4 | Supports this objective by requiring the TSF to destroy key material according to industry standards. |
| | | FCS_COP.1/ENCRYPT | Supports this objective by requiring the TSF to encrypt data according to industry standards |
| | | FCS_COP.1/HASH | Supports this objective by requiring the TSF to hash data according to industry standards. |
| | | FCS_COP.1/SIGN | Supports this objective by requiring the TSF to cryptographically sign data according to industry standards. |
| | | FCS_COP.1/KEYHMAC | Supports this objective by requiring the TSF to perform keyed hashes according to industry standards. |
| | | FIA_X509_EXT.1 | Supports the objective by requiring the TSF to validate certificates using industry standards. |
| | | FIA_X509_EXT.2 | Supports this objective by requiring the TSF to validate TLS and related encrypted connections with x509 certificates. |
| | | FTP_ITC_EXT.1 | Supports the objective by requiring the OS to provide a trusted channel for critical communication. |
| | | FCS_TLS_EXT.1 (TLS Package) | FCS_TLS_EXT.1 supports the objective by defining the TOE's implementation of TLS and DTLS if this protocol is used for protected communications. |

| Source | Objective | Addressed by | Rational |
|---|---|---|---|
| | | FCS_TLSC_EXT.1 (TLS Package) | FCS_TLSC_EXT.1 supports the objective by defining the TOE's implementation of TLS as a client for protected communications. |
| | | FCS_TLSC_EXT.2 (TLS Package) | FCS_TLSC_EXT.2 supports the objective by defining the TOE's implementation of mutually-authenticated TLS as a client for protected communications. |
| | | FCS_TLSC_EXT.5 (TLS Package) | FCS_TLSC_EXT.5 supports the objective be defining the TOE's implementation of supported groups extension for TLS as a client for protected communications. |
| MOD_BT_V1.0 | O.PROTECTED_COMMS | FCS_CKM_EXT.8 | FCS_CKM_EXT.8 supports the objective by requiring the TSF to specify how ECDH key pairs will be refreshed. |
| | | FIA_BLT_EXT.1 | FIA_BLT_EXT.1 supports the objective by ensuring that Bluetooth communications are not initiated without user approval. |
| | | FIA_BLT_EXT.2 | FIA_BLT_EXT.2 supports the objective by requiring the TSF to implement Bluetooth mutual authentication. |
| | | FIA_BLT_EXT.3 | FIA_BLT_EXT.3 supports the objective by preventing Bluetooth spoofing by rejecting connections with duplicate device addresses. |
| | | FIA_BLT_EXT.4 | FIA_BLT_EXT.4 supports the objective by defining the TSF's implementation of Bluetooth Secure Simple Pairing. |
| | | FIA_BLT_EXT.6 | FIA_BLT_EXT.6 supports the objective by requiring the TSF to |

| Source | Objective | Addressed by | Rational |
|---|---|---|---|
| | | | specify the Bluetooth profiles that it requires explicit user authorization to grant access to for trusted devices. |
| | | FIA_BLT_EXT.7 | FIA_BLT_EXT.7 supports the objective by requiring the TSF to specify the Bluetooth profiles that it requires explicit user authorization to grant access to for untrusted devices. |
| | | FTP_BLT_EXT.1 | FTP_BLT_EXT.1 supports the objective by requiring the TSF to implement encryption to protect Bluetooth communications |
| | | FTP_BLT_EXT.2 | FTP_BLT_EXT.2 supports the objective by requiring the TSF to prevent data transmission over Bluetooth if the paired device is not using encryption. |
| | | FTP_BLT_EXT.3/BR | FTP_BLT_EXT.3/BR support the objective by requiring the TSF to implement a minimum encryption key size for Bluetooth BR/EDR. |
| | | FTP_BLT_EXT.3/LE<br>(selection-based) | FTP_BLT_EXT.3/LE support the objective by requiring the TSF to implement a minimum encryption key size for Bluetooth LE. |
| MOD_WLANC_V1.0 | O.AUTH_ COMM | FCS_TLSC_EXT.1/WLAN | FCS_TLSC_EXT.1/WLAN supports the objective by requiring the TSF to use EAP-TLS to establish a secure connection to a wireless access point, including authentication of the access point. |
| | | FIA_PAE_EXT.1 | FIA_PAE_EXT.1 supports the objective by requiring the TSF to act as the supplicant for 802.1X authentication. |

| Source | Objective | Addressed by | Rational |
|---|---|---|---|
| | | FIA_X509_EXT.1/WLAN | FIA_X509_EXT.1/WLAN supports the objective by defining how the TSF determines the validity of presented X.509 certificates. |
| | | FIA_X509_EXT.2/WLAN | FIA_X509_EXT.2/WLAN supports the objective by requiring the TSF to implement X.509 certificate authentication as the mechanism for authentication EAPTLS connections. |
| | | FTP_ITC.1/WLAN | FTP_ITC.1/WLAN supports the objective by requiring the TSF to implement trusted protocols that include authentication of the remote endpoints. |
| | | FCS_TLSC_EXT.2/WLAN<br>(selection-based) | FCS_TLSC_EXT.2/WLAN supports the objective by optionally requiring the TSF to support only certain elliptic curves if the TOE implements any EAP-TLS cipher suites that rely on ECDHE as the key establishment method. |
| MOD_WLANC_V1.0 | O.CRYPTOGRAPHIC_FUNCTIONS | FCS_CKM.1/WPA | FCS_CKM.1/WPA supports the objective by requiring the TSF to generate symmetric keys used for WPA2 and WPA3 in a specified manner. |
| | | FCS_CKM.2/WLAN | FCS_CKM.2/WLAN supports the objective by requiring the TSF to decrypt group temporal keys used for IEEE 802.11. |
| | | FCS_WPA_EXT.1 | FCS_WPA_EXT.1 supports this objective by defining the WPA versions that are supported. |
| MOD_WLANC_V1.0 | O.SELF_ TEST | FPT_TST_EXT.3/WLAN | FPT_TST_EXT.3/WLAN supports the objective by requiring the TSF to perform self-tests to ensure |

| Source | Objective | Addressed by | Rational |
|---|---|---|---|
| | | | that it is operating in a known state. |
| MOD_WLANC_V1.0 | O.SYSTEM_MONITORING | FAU_GEN.1/WLAN | FAU_GEN.1/WLAN supports the objective by requiring the TSF to generate audit records for security-relevant WLAN behavior. |
| MOD_WLANC_V1.0 | O.TOE_ADMINISTRATION | FIA_X509_EXT.6 | FIA_X509_EXT.6 supports the objective by requiring the TSF to securely store certificates in a repository that an administrator can interact with, whether that repository is provided by the WLAN client itself or by a platform storage mechanism defined by the Base-PP portion of the TOE. |
| | | FMT_SMF.1/WLAN | FMT_SMF.1/WLAN supports the objective by requiring the TSF to implement management functionality for security-relevant WLAN behavior. |
| MOD_WLANC_V1.0 | O.WIRELESS_ACCESS_POINT_CONNECTION | FTA_WSE_EXT.1 | FTA_WSE_EXT.1 supports the objective by requiring the TSF to restrict connectivity to allowed wireless networks. |

# 7 Appendix

## 7.1 SFR to CAVP Mapping Table

187 The CAVP certificates contain several different SoCs and micro-architectures in the operational environment (OE). The relationship between the SoCs and micro-architectures used by the devices claimed in this evaluation are specified in Table 5.

188 The following conventions are used in the tables of this appendix to identify the cryptographic modules leveraged by the TOE.

a) **KRN.** Apple corecrypto Module v14.0 [Apple Silicon, Kernel, Software, SL1]

b) **SEP.** Secure Enclave Processor Hardware v2.0 onboard Apple Silicon

c) **SKS.** Apple corecrypto Module v14.0 [Apple Silicon, Secure Key Store, Hardware, SL2]

d) **USR.** Apple corecrypto Module v14.0 [Apple Silicon, User, Software, SL1]

e) **BC.** Crypto Hardware Module aes_core_gcm.vhd

**Table 23: Cryptographic Algorithm Table- Apple Silicon**

| SFR | Algorithm | Capabilities | Mod | Implementation | CAVP |
|---|---|---|---|---|---|
| FCS_CKM.1 | RSA KeyGen [FIPS PUB 186-4] | 3072-bits, 4096-bits | USR | vng_ltc | A5988 |
| | ECDSA KeyGen [FIPS PUB 186-4] | P-256, P-384, P-521 | USR | vng_ltc | A5988 |
| FCS_CKM.2 | RSA Key Establishment [RFC 8017] | 3072-bits, 4096-bits | USR | c_ltc | Tested by the CCTL per the testing Assurance Activity for FCS_CKM.2 |
| | ECC Key Establishment (KAS-ECC-SSCSp800-56Ar3)<br><br>[SP800-56A-Rev3] | P-256, P-384, P-521 | USR | c_ltc | A5986 |
| FCS_CKM.2/WLAN | AES-KW | 256-bit | USR | c_ltc | A5986 |
| | AES-KW | 256-bit | USR | c_ltc | A5986 |

| SFR | Algorithm | Capabilities | Mod | Implementation | CAVP |
|---|---|---|---|---|---|
| FCS_COP.1/ ENCRYPT | AES-CTR | 256-bit | BC | 4388 | AES 5926 |
| | | | | 4387 | AES 5926 |
| | | | | 4378 | AES 5926 |
| | AES-GCM | 256-bit | USR | vng_asm | A5987 |
| | AES-CCM | 128-bit | BC | 4388 | AES 5926 |
| | | | | 4387 | AES 5926 |
| | | | | 4378 | AES 5926 |
| | AES-GCMP-256 | 256-bit | BC | 4388 | AES 5926 |
| | | | | 4387 | AES 5926 |
| | | | | 4378 | AES 5926 |
| FCS_COP.1/HASH | SHS Byte-oriented mode | SHA-256, SHA-384, SHA-512 | USR | vng_ltc | A5988 |
| | | SHA-256 | KRN | vng_ltc | A5949 |
| | | | SKS | vng_ltc | A5949 |
| FCS_COP.1/SIGN | RSA SigGen/SigVer | Modulo: 3072-bits, 4096-bits<br>with: SHA-256, SHA-384, SHA-512 | USR | vng_ltc | A5988 |
| | ECDSA SigGen/SigVer | P-384, P-521 with: SHA-256, SHA-384, SHA-512 | USR | vng_ltc | A5988 |
| | | P-521 with: SHA-512 | KRN | vng_ltc | A5949 |
| | | | SKS | vng_ltc | A5949 |
| FCS_COP.1/KEYHMAC | HMAC Byte-oriented mode | HMAC-SHA-256, HMAC-SHA-384, | USR | vng_ltc | A5988 |
| FCS_RBG_EXT.1 | CTR_DRBG | AES-256 | USR | vng_asm | A5987 |

| SFR | Algorithm | Capabilities | Mod | Implementation | CAVP |
|---|---|---|---|---|---|
| | | | SEP | M2 Max (trng)<br>M2 Pro (trng)<br>M2 (trng)<br>M1 Ultra (trng)<br>M1 Max (trng)<br>M1 Pro (trng) | A3490 |
| | | | | M1 (trng) | A1362 |

189 In addition, the following table is provided to show conformance to NIAP Policy 5.

**Table 24: NIAP Policy 5 Mapping**

| SFR | Cryptographic Operation | NIST Standard | CAVP Certificate (Algorithm) |
|---|---|---|---|
| FCS_CKM.1 | Asymmetric Key Generation:<br>RSA schemes using cryptographic key sizes of 3072-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 | FIPS PUB 186-4 | A5988 (RSA) |
| FCS_CKM.1 | Asymmetric Key Generation:<br>ECC schemes using "NIST curves" P-384 and [P-256, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 | FIPS PUB 186-4 | A5988 (ECDSA) |
| FCS_CKM.2 | RSA-based key establishment schemes that meets the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.2" | N/A | CCTL Tested |
| FCS_CKM.2 | Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” | NIST SP 800-56A Revision 3 | A5986 (KAS-ECC-SSCSp800-56Ar3) |

| SFR | Cryptographic Operation | NIST Standard | CAVP Certificate (Algorithm) |
|---|---|---|---|
| FCS_CKM.2/WPA | The TSF shall generate symmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [PRF-384 and [PRF-704] (as defined in IEEE 802.11- 2012)] and specified key sizes [256 bits and [no other key sizes]] using a Random Bit Generator as specified in FCS_RBG_EXT.1. | FIPS PUB 197<br>NIST SP 800-38F | A5986 (AES-KW) |
| FCS_COP.1/ ENCRYPT | encryption/decryption services for data in accordance with a specified cryptographic algorithm [:<br>AES-CTR (as defined in NIST SP 800-38A)<br>AES Key Wrap (KW) (as defined in NIST SP 800-38F)<br>AES-GCMP-256 (as defined in NIST SP 800-38D and IEEE 802.11ac-2013)<br>AES-CCM (as defined in NIST SP 800-38C)<br>AES-GCM (as defined in NIST SP 800-38D)<br>] and cryptographic key sizes 256-bit and [128-bit]] | FIPS PUB 197<br>NIST SP 800-38A<br>NIST SP 800-38C<br>NIST SP 800-38D<br>NIST SP 800-38F | A5986 (AES-KW)<br>AES 5926 (AES-CTR)<br>A5987 (AES-GCM)<br>AES 5926 (AES-CCM)<br>AES 5926 (AES-GCMP) |
| FCS_COP.1/ HASH | cryptographic hashing services in accordance with a specified cryptographic algorithm [SHA-256, SHA-384, SHA-512] | FIPS PUB 180-4 | A5988 (SHA-256, SHA-384, SHA-512)<br>A5949 (SHA-256) |
| FCS_COP.1/ SIGN | RSA schemes using cryptographic key sizes of [3072-bit or greater] that meet the following: FIPS PUB 186-4 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 4 | FIPS PUB 186-4 | A5988 (RSA) |
| FCS_COP.1/ SIGN | ECDSA schemes using "NIST curves" P-384 and [P-521] that meet the following: FIPS PUB 186-4 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5, SP 800-186 | FIPS PUB 186-4 | A5988 (ECDSA)<br>A5949 |

| SFR | Cryptographic Operation | NIST Standard | CAVP Certificate (Algorithm) |
|---|---|---|---|
| FCS_COP.1/ KEYHMAC | keyed-hash message authentication services in accordance with a specified cryptographic algorithm [HMAC-SHA-256, HMAC-SHA-384]...used in HMAC] | FIPS PUB 198-1 | A5988 (HMAC-SHA-256, HMAC-SHA-384) |
| FCS_RBG_EXT.1 | deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [<br>CTR_DRBG (AES)<br>] | NIST SP 800-90A | A5987 (CTR_DRBG)<br>A3490 (CTR_DRBG)<br>A1362 (CTR_DRBG) |