Apple macOS 15 Sequoia Security Target

Version:	1.2
Status:	Final
Date:	2026-02-10
Validation ID:	11648
Classification:	Public
Prepared for:	Apple Inc.
Prepared by:	atsec information security corporation

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

Legal Notice

This document is provided AS IS with no express or implied warranties. Use the information in this document at your own risk.

This document may be reproduced or distributed in any form without prior permission provided the copyright notice is retained on all copies. Modified versions of this document may be freely distributed provided that they are clearly identified as such, and this copyright is included intact.

Revision History

Version	Date	Author(s)	Changes to Previous Revision
1.0	2025-08-29	atsec	First version
1.1	2025-12-09	atsec	Updated as per comments from evaluator and vendor. Added TD0958.
1.2	2026-02-10	atsec	Fixed typos in Table 3. Address ECR comments

Table of Contents

List of Tables

1 Introduction

1.1 Security Target Identification

Title:	Apple macOS 15 Sequoia Security Target
Version:	1.2
Status:	Final
Date:	2026-02-10
Sponsor:	Apple Inc.
Developer:	Apple Inc.
Validation Body:	NIAP
Validation ID:	11648
Keywords:	Operating System, macOS, Apple silicon

1.2 TOE Identification

The TOE is Apple macOS 15 Sequoia.

1.3 TOE Type

The TOE type is a general-purpose operating system and provides wireless LAN and Bluetooth functionality.

1.4 TOE Overview

The Target of Evaluation (TOE) is Apple macOS 15 Sequoia, which is a general purpose operating system running on the Apple Mac computers with Apple silicon.

The TOE is a Unix-based operating system built on top of the XNU kernel. The TOE implements standard Unix facilities, provides both command-line and graphical user interfaces, supports Bluetooth communication, and includes Wireless Local Area Network (WLAN) client functionality. A portion of the Bluetooth and WLAN functionalities is implemented in hardware (Broadcom chip).

The tested version of the TOE is macOS 15.3.

1.5 TOE Description

This section provides a general description of the TOE, including architecture, physical boundaries, security functions, and relevant TOE documentation.

1.5.1 Architecture

The TOE is a general purpose operating system running on the Apple Mac computers with Apple silicon. The Mac computers covered in this evaluation contain the Secure Enclave, a dedicated secure subsystem integrated into the Apple silicon.

Utilizing a dedicated processor (Secure Enclave Processor or SEP), the Secure Enclave is isolated from the main processor to provide an extra layer of security designed to keep sensitive data secure. The SEP executes the SEP Operating System (sepOS), which is based on a customized version of the L4 microkernel. The sepOS is included with macOS and is within the TOE boundary. The Secure Enclave supports the TOE for secure boot, and the generation of secure random data used in cryptographic key generation.

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.

The TOE hardware platforms include a third-party chip that implements Bluetooth and wireless LAN functionality. The chip model depends on the hardware platforms listed in Appendix A.

1.5.2 TOE Physical Boundary

The physical boundary of the TOE is the installation image which includes both macOS and sepOS. The TOE hardware platforms covered by this evaluation are listed in Appendix A.

1.5.3 TOE Security Functionality

The TOE provides the security functions required by the conformance claims defined in Section 2.

1.5.3.1 Security Audit

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 Section 2. Audit events are generated for the following audit functions:

- 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)

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.

1.5.3.2 Cryptographic Support

The TOE includes the Apple corecrypto v18.3 cryptographic implementations and is supported by the onboard Apple Secure Enclave hardware for performing user space, kernel space, and Secure Enclave cryptographic operations. In addition, it uses software and hardware noise sources for entropy generation.

The TOE implements Transport Layer Security version 1.2 (TLS 1.2) for secure communications with remote servers. The TOE claims conformance to Functional Package for Transport Layer Security v1.1 (PKG_TLS_V1.1), which does not cover TLS version 1.3. Therefore, TLS version 1.3 is out of scope of the evaluation, and this document focuses on TLS version 1.2.

The Bluetooth hardware implements the AES-CCM-128 cryptographic functionality used when connecting to remote Bluetooth devices. The TOE implements Wi-Fi Protected Access (WPA2 and WPA3) to secure 802.11 wireless traffic protected using AES-CCMP-256 and AES-GCMP-256 cryptographic algorithms.

Relevant Cryptographic Algorithm Validation Program (CAVP) certificates are shown in Section 7.1.2.

1.5.3.3 User Data Protection

The TOE implements access controls that prevent unprivileged users from accessing files and directories owned by other users.

1.5.3.4 Identification and Authentication

All users must be authenticated to the TOE prior to carrying out any management actions. The TOE supports:

- Password-based authentication.
- Authentication based on username and a PIN that releases the asymmetric key stored in OE-protected storage.

The TOE will deny further user authentication once a defined number of unsuccessful authentication attempts have been reached.

For Bluetooth, the TOE supports 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 already exists. The TOE requires explicit user authorization when pairing with an untrusted device.

External entities connecting to the TOE via a secure protocol (e.g., TLS, Extensible Authentication Protocol TLS (EAP-TLS)) can be authenticated using X.509 certificates.

1.5.3.5 Security Management

The TOE can perform management functions. The administrator has full access to carry out all management functions, whereas the user has limited privileges.

1.5.3.6 Protection of the TSF

The TOE implements the following protection of TOE Security Functionality:

- Access controls for critical components.
- Address space layout randomization (ASLR) with 16 bits of entropy.
- Stack buffer overflow protection.
- Verification of integrity of the bootchain and operating system executable code.
- Trusted software updates using digital signatures.

1.5.3.7 TOE Access

Before establishing a user session, the TOE can display an advisory warning message regarding unauthorized use of the OS. Access to the TOE via a wireless network is controlled by administrator defined policy.

1.5.3.8 Trusted Path/Channels

The TOE supports TLS 1.2 for trusted channel communications. The TOE uses TLS to securely communicate with the Apple Update Server. Applications may invoke the TOE-provided TLS to securely communicate with remote servers. The TOE enforces encryption when transmitting data over Bluetooth and terminates the connection if the connected device stops encrypting. The TOE uses EAP-TLS for authentication and WPA for data encryption when connecting to a wireless access point as the WLAN client.

1.5.4 TOE Guidance

The following TOE guidance document in PDF format is publicly available on the NIAP Product Compliance List (PCL) alongside this Security Target:

Apple macOS 15 Sequoia Common Criteria Guide (CCGUIDE)

1.5.5 TOE Operational Environment

The following environmental components interoperate with the TOE in the evaluated configuration.

Table 1: TOE Operational Environment

Component	Description
Hardware platform	Apple Mac computers listed in Appendix A

Apple Update Server	Server that allows the TOE to download updates
External TLS server (optional)	Applications on the TOE may initiate TLS sessions to external TLS servers

2 CC Conformance Claim

This Security Target (ST) is CC Part 2 extended and CC Part 3 extended. Common Criteria (CC) version 3.1 revision 5 is the basis for this conformance claim.

This ST claims exact conformance to the following Protection Profiles (PPs) and Functional Packages:

- PP-Configuration for General Purpose Operating System, Bluetooth, and Wireless Local Area Network Clients, Version 1.0 (CFG_GPOS_BT_WLANC_V1.0).

  This PP-Configuration includes the following components:

  - Base-PP: Protection Profile for General Purpose Operating Systems, Version 4.3 (PP_OS_V4.3);
  - PP-Module: PP-Module for Bluetooth, Version 1.0 (MOD_BT_V1.0); and
  - PP-Module: PP-Module for WLAN Clients, Version 1.0 (MOD_WLANC_V1.0).
- Functional Package for Transport Layer Security (TLS), Version 1.1 (PKG_TLS_V1.1).

The following table contains the NIAP Technical Decisions (TDs) for the Protection Profile for General Purpose Operating Systems, Version 4.3 (PP_OS_V4.3) at the time of the evaluation and a statement of applicability to the evaluation.

Table 2: NIAP Technical Decisions for PP_OS_V4.3

NIAP TD	Description	Applicable?	Non-applicability Rationale
TD0958	Correction to Referenced PPs in FTP_ITC_EXT.1	Yes	
TD0955	Adding FIPS 186-5 in PP_OS_V4.3	Yes	
TD0930	Clarification when CTR_DRBG is Selected for FCS_RBG_EXT.1.2 in PP_OS_V4.3	Yes	
TD0914	Addition of PKG_TLS_V2.0 to Conformance Claims	No	An exemption was granted by NIAP.
TD0906	Clarification to List of Examples in FPT_SBOP_EXT.1	Yes	
TD0904	Addition of MOD_VPNC_V2.5 to Conformance Claims	No	The ST does not claim MOD_VPNC_V2.5.
TD0844	Addition of Assurance Package for Flaw Remediation V1.0 Conformance Claim	No	The ST does not claim Assurance Package for Flaw Remediation V1.0.
TD0839	Clarification for Local Administration in FTP_TRP.1.3	Yes	
TD0821	Corrections to ECD for PP_OS_V4.3	Yes	
TD0812	Updated CC Conformance Claims in PP_OS_v4.3	Yes	
TD0789	Correction to TLS Selection in FIA_X509_EXT.2.1	Yes	
TD0773	Updates to FIA_X509_EXT.1 for Exception Processing and Test Conditions	Yes	
TD0713	Functional Package SFR mappings to objectives	Yes	

TD0712	Support for Bluetooth Standard 5.3	Yes	
TD0701	Incomplete selection reference in FCS_CKM_EXT.4 TSS activities	Yes	
TD0696	Removal of 160 bit selection from FCS_COP.1/HASH & FCS_COP.1/KEYHMAC	Yes	
TD0693	Typos in OSPP 4.3	Yes	
TD0691	OSPP 4.3 Conditional authentication testing	Yes	
TD0675	Make FPT_W^X_EXT.1 Optional	Yes	

The following table contains the TDs for the PP-Module for Bluetooth, Version 1.0 (MOD_BT_V1.0) at the time of the evaluation and a statement of applicability to the evaluation.

Table 3: NIAP Technical Decisions for MOD_BT_V1.0

NIAP TD	Description	Applicable?	Non-applicability Rationale
TD0707	Formatting corrections for MOD_BT_V1.0	Yes	
TD0685	BT missing multiple SFR-to-Obj mappings	Yes	
TD0671	Bluetooth PP-Module updated to allow for new PP and PP-Module Versions	Yes	
TD0650	Conformance claim sections updated to allow for MOD_VPNC_V2.3 and 2.4	No	This evaluation does not include MOD_VPNC_V2.3 or V2.4.
TD0645	Bluetooth audit details	Yes	
TD0640	Handling BT devices that do not support encryption	Yes	
TD0600	Conformance claim sections updated to allow for MOD_VPNC_V2.3	No	This evaluation does not include MOD_VPNC_V2.3.

The following table contains the TDs for the PP-Module for WLAN Clients, Version 1.0 (MOD_WLANC_V1.0) at the time of the evaluation and a statement of applicability to the evaluation.

Table 4: NIAP Technical Decisions for MOD_WLANC_V1.0

NIAP TD	Description	Applicable?	Non-applicability Rationale
TD0920	Clarification for FMT_SMF.1/WLAN Table 3	Yes	
TD0837	Updates to WLAN Client PP-Module allow-lists	Yes	
TD0797	Addition of FCS_WPA_EXT to ECD	Yes	
TD0710	WPA version restrictions	Yes	
TD0703	Removal of FIA_X509_EXT.2/WLAN evaluation activities for revocation checking	Yes	

TD0667	Move Set Wireless Freq Band to Optional/Objective	Yes	

The following table contains the TDs for the Functional Package for Transport Layer Security (TLS), Version 1.1 (PKG_TLS_V1.1) at the time of the evaluation and a statement of applicability to the evaluation.

Table 5: NIAP Technical Decisions for PKG_TLS_V1.1

NIAP TD	Description	Applicable?	Non-applicability Rationale
TD0779	Updated Session Resumption Support in TLS package V1.1	No	The TOE does not contain TLS server functionality.
TD0770	TLSS.2 connection with no client cert	No	The TOE does not contain TLS server functionality.
TD0739	PKG_TLS_V1.1 has 2 different publication dates	Yes	
TD0726	Corrections to (D)TLSS SFRs in TLS 1.1 FP	No	The TOE does not contain (D)TLS server functionality.
TD0513	CA Certificate loading	Yes	
TD0499	Testing with pinned certificates	Yes	
TD0469	Modification of test activity for FCS_TLSS_EXT.1.1 test 4.1	No	The TOE does not contain TLS server functionality.
TD0442	Updated TLS Ciphersuites for TLS Package	Yes	

3 Security Problem Definition

The threats, assumptions, and organizational security policies (OSPs) are defined in the documents specified in Section 2 "CC Conformance Claim". This Security Target includes by reference the Security Problem Definition (composed of threats, assumptions, and organizational security policies) from PP_OS_V4.3, MOD_BT_V1.0, and MOD_WLANC_V1.0.

4 Security Objectives

The Security Objectives for the TOE and Security Objectives for the Operational Environment are defined in the documents specified in Section 2 "CC Conformance Claim". This Security Target includes by reference the Security Objectives from PP_OS_V4.3, MOD_BT_V1.0, and MOD_WLANC_V1.0.

5 Extended Components Definition

All extended SFR components are defined in the PP/PP-Module/Package claimed in Section 2.

One extended SAR component, ALC_TSU_EXT.1 Timely Security Updates, is defined in PP_OS_V4.3.

6 Security Requirements

6.1 Security Functional Requirements

The following conventions are used to indicate the operations performed within the ST on security requirement components:

- Selections are shown in **bold text** and are surrounded by square brackets.
- Assignments are shown in *italic text* and are surrounded by square brackets. The assignments within a selection are shown in ***bold italics***.
- Iterations are identified by appending a suffix to the original SFR.
- Refinements added to the text are shown in underlined text, deletions are shown as ~~strikethrough text~~.

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 - Audit Data Generation (Refined)

Origin: PP_OS_V4.3

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*].

TSS Link: Section 7.1.1

6.1.1.2 FAU_GEN.1/BT – Audit Data Generation (Bluetooth)

Origin: MOD_BT_V1.0

Applied TDs: TD0707, TD0645

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.

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.

TSS Link: Section 7.1.1

Table 6: Auditable Events (Bluetooth)

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**].
	Failure of Bluetooth connection.	Reason for failure.
~~FIA_BLT_EXT.3 (optional)~~	~~Duplicate connection attempt.~~	~~[selection: complete, last [assignment: integer greater than or equal to 2] octets of the] BD_ADDR of connection attempt~~
FIA_BLT_EXT.4	None.	
~~FIA_BLT_EXT.5~~	~~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	None.	

Application Note:

FIA_BLT_EXT.3 is crossed out because the rejection is performed at the HCI layer.

FIA_BLT_EXT.5 is crossed out because it is not claimed in the ST.

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

Origin: MOD_WLANC_V1.0

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 7 and selected SFRs in ~~Table 5~~ Table 8.

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 7 and selected SFRs in ~~Table 5~~ Table 8.

TSS Link: Section 7.1.1

Table 7: Auditable Events for Mandatory Requirements (WLAN)

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. • Non-TOE endpoint of connection.
	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.
	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.

	[**None**].	[**None**].
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 - 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 8: Auditable Events for Selection-based Requirements (WLAN)

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

6.1.2 FCS – Cryptographic Support

6.1.2.1 FCS_CKM.1 – Cryptographic Key Generation (Refined)

Origin: PP_OS_V4.3

Applied TDs: TD0712, TD0955

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 186-5, "Digital Signature Standard (DSS)", Appendix A.1**
- **ECC schemes using "NIST curves" P-384 and [P-256, P-521] that meet the following: FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2**

].

TSS Link: Section 7.1.2.1

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

Origin: MOD_WLANC_V1.0

FCS_CKM.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.

TSS Link: Section 7.1.2.1

6.1.2.3 FCS_CKM.2 - Cryptographic Key Establishment (Refined)

Origin: PP_OS_V4.3

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"**

].

TSS Link: Section 7.1.2.2

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

Origin: MOD_WLANC_V1.0

FCS_CKM.2/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.

TSS Link: Section 7.1.2.3

6.1.2.5 FCS_CKM_EXT.4 - Cryptographic Key Destruction

Origin: PP_OS_V4.3

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 [**
  - **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.

TSS Link: Section 7.1.2.4

6.1.2.6 FCS_CKM_EXT.8 - Bluetooth Key Generation

Origin: MOD_BT_V1.0

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

TSS Link: Section 7.1.2.5

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

Origin: PP_OS_V4.3

Applied TDs: TD0712

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-CCMP-256 (as defined in NIST SP 800-38C and IEEE 802.11ac-2013)**
- **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**].

TSS Link: Section 7.1.2.6

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

Origin: PP_OS_V4.3

Applied TDs: TD0696

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.

TSS Link: Section 7.1.2.7

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

Origin: PP_OS_V4.3

Applied TDs: TD0955

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**

].

TSS Link: Section 7.1.2.8

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

Origin: PP_OS_V4.3

Applied TDs: TD0696

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.

TSS Link: Section 7.1.2.9

6.1.2.11 FCS_RBG_EXT.1 - Random Bit Generation

Origin: PP_OS_V4.3

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.

TSS Link: Section 7.1.2.10

6.1.2.12 FCS_STO_EXT.1 - Storage of Sensitive Data

Origin: PP_OS_V4.3

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.

TSS Link: Section 7.1.2.11

6.1.2.13 FCS_TLS_EXT.1 - TLS Protocol

Origin: PKG_TLS_V1.1

FCS_TLS_EXT.1.1 The product shall implement [

- **TLS as a client**

].

TSS Link: Section 7.1.2.12

6.1.2.14 FCS_TLSC_EXT.1 - TLS Client Protocol

Origin: PKG_TLS_V1.1

Applied TDs: TD0442

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**
- **session renegotiation**

].

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**

].

TSS Link: Section 7.1.2.12

6.1.2.15 FCS_TLSC_EXT.1/WLAN - TLS Client Protocol (EAP-TLS for WLAN)

Origin: MOD_WLANC_V1.0

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_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**

].

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.

TSS Link: Section 7.1.2.13

6.1.2.16 FCS_TLSC_EXT.2 - TLS Client Support for Mutual Authentication

Origin: PKG_TLS_V1.1

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

TSS Link: Section 7.1.2.12

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

Origin: MOD_WLANC_V1.0

FCS_TLSC_EXT.2.1/ WLAN The TSF shall present the Supported Groups extension in the Client Hello with the following NIST curves: [**secp384r1, secp521r1**].

TSS Link: Section 7.1.2.13

6.1.2.18 FCS_TLSC_EXT.4 - TLS Client Support for Renegotiation

Origin: PKG_TLS_V1.1

FCS_TLSC_EXT.4.1 The product shall support secure renegotiation through use of the “renegotiation_info” TLS extension in accordance with RFC 5746.

TSS Link: Section 7.1.2.12

6.1.2.19 FCS_TLSC_EXT.5 - TLS Client Support for Supported Groups Extension

Origin: PKG_TLS_V1.1

FCS_TLSC_EXT.5.1 The product shall present the Supported Groups Extension in the Client Hello with the supported groups [

- **secp384r1,**
- **secp521r1,**

].

TSS Link: Section 7.1.2.12

6.1.2.20 FCS_WPA_EXT.1 - Supported WPA Versions

Origin: MOD_WLANC_V1.0

FCS_WPA_EXT.1.1 The TSF shall support WPA3 and [**WPA2**] security type.

TSS Link: Section 7.1.2.14

6.1.3 FDP – User Data Protection

6.1.3.1 FDP_ACF_EXT.1 - Access Controls for Protecting User Data

Origin: PP_OS_V4.3

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.

TSS Link: Section 7.1.3

6.1.4 FIA - Identification and Authentication (FIA)

6.1.4.1 FIA_AFL.1 - Authentication failure handling (Refined)

Origin: PP_OS_V4.3

Applied TDs: TD0691

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: [*deny subsequent authentication attempts*].

TSS Link: Section 7.1.4.1

6.1.4.2 FIA_BLT_EXT.1 - Bluetooth User Authorization

Origin: MOD_BT_V1.0

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

TSS Link: Section 7.1.4.2

6.1.4.3 FIA_BLT_EXT.2 - Bluetooth Mutual Authentication

Origin: MOD_BT_V1.0

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

TSS Link: Section 7.1.4.3

6.1.4.4 FIA_BLT_EXT.3 - Rejection of Duplicate Bluetooth Connections

Origin: MOD_BT_V1.0

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.

TSS Link: Section 7.1.4.4

6.1.4.5 FIA_BLT_EXT.4 - Secure Simple Pairing

Origin: MOD_BT_V1.0

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.

TSS Link: Section 7.1.4.5

6.1.4.6 FIA_BLT_EXT.6 - Trusted Bluetooth Device User Authorization

Origin: MOD_BT_V1.0

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*].

TSS Link: Section 7.1.4.6

6.1.4.7 FIA_BLT_EXT.7 - Untrusted Bluetooth Device User Authorization

Origin: MOD_BT_V1.0

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: [*all*].

TSS Link: Section 7.1.4.6

6.1.4.8 FIA_PAE_EXT.1 - Port Access Entity Authentication

Origin: MOD_WLANC_V1.0

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

TSS Link: Section 7.1.4.7

6.1.4.9 FIA_UAU.5 - Multiple Authentication Mechanisms (Refined)

Origin: PP_OS_V4.3

Applied TDs: TD0691

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*].

TSS Link: Section 7.1.4.8

6.1.4.10 FIA_X509_EXT.1 - X.509 Certificate Validation

Origin: PP_OS_V4.3

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.

TSS Link: Section 7.1.4.9

6.1.4.11 FIA_X509_EXT.1/WLAN - X.509 Certificate Validation

Origin: MOD_WLANC_V1.0

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.

TSS Link: Section 7.1.4.9

6.1.4.12 FIA_X509_EXT.2 - X.509 Certificate Authentication

Origin: PP_OS_V4.3

Applied TDs: TD0789

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

TSS Link: Section 7.1.4.10

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

Origin: MOD_WLANC_V1.0

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.

TSS Link: Section 7.1.4.10

6.1.4.14 FIA_X509_EXT.6 - X.509 Certificate Storage and Management

Origin: MOD_WLANC_V1.0

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.

TSS Link: Section 7.1.4.11

6.1.5 FMT - Security Management (FMT)

6.1.5.1 FMT_MOF_EXT.1 - Management of Security Functions Behavior

Origin: PP_OS_V4.3

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.

TSS Link: Section 7.1.5

6.1.5.2 FMT_MOF_EXT.1/BT - Management of Security Functions Behavior

Origin: MOD_BT_V1.0

FMT_SMF_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.

TSS Link: Section 7.1.5

6.1.5.3 FMT_SMF_EXT.1 - Specification of Management Functions

Origin: PP_OS_V4.3

Applied TDs: TD0693

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

Table 9: 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	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	M
19	Enable/disable Bluetooth interface	M	M
20	Enable/disable [*no other external interfaces*]	-	-
21	[*No other management functions to be provided by the TSF*]	-	-

Application Note:

M: Supported by the specified role.

Grey/hyphen: Not supported by the specified role.

The use of ‘M’ and ‘-‘ as indicator markers follows TD0693.

TSS Link: Section 7.1.5

6.1.5.4 FMT_SMF_EXT.1/BT - Specification of Management Functions

Origin: MOD_BT_V1.0

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

Table 10: Management Functions (Bluetooth)

Function	**Administrator**	**User**
BT-1. Configure the Bluetooth trusted channel. • Disable/enable the Discoverable (for BR/EDR) and Advertising (for LE) modes;	X	X
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: [selection: Wi-Fi, NFC, [assignment: other wireless technologies]];	-	-
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.

Grey/hyphen: Not supported by the specified role.

TSS Link: Section 7.1.5

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

Origin: MOD_WLANC_V1.0

Applied TDs: TD0667, TD0920

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

TSS Link: Section 7.1.5

Table 11: Management Functions (WLAN)

#	Management Function	Impl	Admin	User
WL-1	configure security policy for each wireless network: • [**specify the CA(s) from which the TSF will accept WLAN authentication server certificate(s)**], • security type, • authentication protocol, • client credentials to be used for authentication	X	X	-
WL-2	specify wireless networks (SSIDs) to which the TSF may connect	X	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**]	X	X	-
WL-4	enable/disable certificate revocation list checking	-	-	-
WL-5	disable ad hoc wireless client-to-client connection capability (a.k.a. Apple AirDrop)	X	X	X
WL-6	disable roaming capability	-	-	-
WL-7	enable/disable IEEE 802.1X pre-authentication	-	-	-
WL-8	loading X.509 certificates into the TOE	X	X	-
WL-9	revoke X.509 certificates loaded into the TOE	X	X	-
WL-10	enable/disable and configure PMK caching: • set the amount of time (in minutes) for which PMK entries are cached • set the maximum number of PMK entries that can be cached	-	-	-
WL-11	configure security policy for each wireless network: set wireless frequency band to [**2.4 GHz, 5 GHz, 6 GHz**]	-	-	-

Application Note:

X: Supported by the specified role or implemented in the TOE.

Grey/hyphen: Not supported by the specified role or not implemented in the TOE.

6.1.6 FPT – Protection of the TSF

6.1.6.1 FPT_ACF_EXT.1 - Access Controls

Origin: PP_OS_V4.3

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]*.

TSS Link: Section 7.1.6.1

6.1.6.2 FPT_ASLR_EXT.1 - Address Space Layout Randomization

Origin: PP_OS_V4.3

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

TSS Link: Section 7.1.6.2

6.1.6.3 FPT_SBOP_EXT.1 - Stack Buffer Overflow Protection

Origin: PP_OS_V4.3

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

TSS Link: Section 7.1.6.3

6.1.6.4 FPT_TST_EXT.1 – Boot Integrity

Origin: PP_OS_V4.3

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**

] .

TSS Link: Section 7.1.6.4

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

Origin: MOD_WLANC_V1.0

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.

TSS Link: Section 7.1.6.4

6.1.6.6 FPT_TUD_EXT.1 - Trusted Update

Origin: PP_OS_V4.3

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.

TSS Link: Section 7.1.6.5

6.1.6.7 FPT_TUD_EXT.2 - Trusted Update for Application Software

Origin: PP_OS_V4.3

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.

TSS Link: Section 7.1.6.5

6.1.7 FTA – TOE Access

6.1.7.1 FTA_TAB.1 - Default TOE Access Banners

Origin: PP_OS_V4.3

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

TSS Link: Section 7.1.7.1

6.1.7.2 FTA_WSE_EXT.1 - Wireless Network Access

Origin: MOD_WLANC_V1.0

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.

TSS Link: Section 7.1.7.2

6.1.8 FTP – Trusted Path/Channel

6.1.8.1 FTP_BLT_EXT.1 - Bluetooth Encryption

Origin: MOD_BT_V1.0

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.

TSS Link: Section 7.1.8.1

6.1.8.2 FTP_BLT_EXT.2 - Persistence of Bluetooth Encryption

Origin: MOD_BT_V1.0

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

TSS Link: Section 7.1.8.2

6.1.8.3 FTP_BLT_EXT.3/BR - Bluetooth Encryption Parameters (BR/EDR)

Origin: MOD_BT_V1.0

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.

TSS Link: Section 7.1.8.3

6.1.8.4 FTP_BLT_EXT.3/LE - Bluetooth Encryption Parameters (LE)

Origin: MOD_BT_V1.0

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.

TSS Link: Section 7.1.8.3

6.1.8.5 FTP_ITC_EXT.1 - Trusted Channel Communication

Origin: PP_OS_V4.3

Applied TDs: TD0789, TD0958

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: [***Apple Update Server, application initiated TLS***] 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.

TSS Link: Section 7.1.8.4

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

Origin: MOD_WLANC_V1.0

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.

TSS Link: Section 7.1.8.4

6.1.8.7 FTP_TRP.1 - Trusted Path

Origin: PP_OS_V4.3

Applied TDs: TD0839

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**].

TSS Link: Section 7.1.8.5

6.2 Security Assurance Requirements

The Security Assurance Requirements (SARs) are included by reference from PP_OS_V4.3. The table below summarizes the SARs for TOE evaluation.

Table 12: Assurance Requirements

Requirement Class	**Requirement Components**
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives (ASE_OBJ.2)
	Stated security requirements (ASE_REQ.2)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance Documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life-cycle Support (ALC)	Labelling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
	Timely security updates (ALC_TSU_EXT.1)
Tests (ATE)	Independent testing – conformance (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability survey (AVA_VAN.1)

7 TOE Summary Specification

7.1 TOE Security Functionality

As per the conformance claims defined in Section 2, the TOE supports the following security features:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

7.1.1 Security Audit (FAU)

The TOE generates audit records using two logging subsystems: auditd and Apple Unified Logging. In the evaluated configuration, the TOE employs auditd to log the auditable events specified in OS_PP_V4.3 and uses the Apple Unified logging to generate Bluetooth and WLAN audit records. The TOE includes the built-in utilities praudit and auditreduce for accessing audit records generated by auditd, and the log utility for accessing audit records generated by the Apple Unified Logging system.

The praudit command is used to make the binary output of the auditreduce command readable. The praudit command reads audit records in binary format from standard input and displays the records in multiple formats, such as XML.

The audit record XML format is:

```
<record><argument/><path/><attribute/><subject/><text/><return/><identity/></record>
```

The <record> tag specifies the beginning and end of an audit record. The <argument> tag specifies the command line arguments. The <path> tag indicates the filesystem object path name. The <attribute> tag indicates the subject (e.g., user) trigger the audit event. The <subject> tag indicates the subject (e.g., user) triggering the audit event. The <text> tag contains the audit event description. The <return> tag specifies the outcome of the audited event. The <identity> tag indicates the code-signing information of the component generating the audit event.

An audit record generated by the Apple Unified Logging system looks like this:

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

The timestamp indicates 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 includes specific information about the event, such as related processes, subsystems, and libraries, along with a human-readable description.

7.1.2 Cryptographic Support (FCS)

The TOE uses the following cryptographic implementations:

- USR: Apple corecrypto Module v18.3 [Apple silicon, User, Software, SL1]
- KRN: Apple corecrypto Module v18.3 [Apple silicon, Kernel, Software, SL1]
- SKS: Apple corecrypto Module v18.3 [Apple silicon, Secure Key Store, Hardware, SL2]

- SE: Secure Enclave hardware onboard Apple silicon
- BC: Broadcom Crypto Hardware Module

The USR, KRN, and SKS cryptographic implementations execute on the following operational environments:

- macOS Sequoia 15 on Apple M Series (ARMv8.5-A) M1
- macOS Sequoia 15 on Apple M Series (ARMv8.5-A) M1 Pro
- macOS Sequoia 15 on Apple M Series (ARMv8.5-A) M1 Max
- macOS Sequoia 15 on Apple M Series (ARMv8.5-A) M1 Ultra
- macOS Sequoia 15 on Apple M Series (ARMv8.6-A) M2
- macOS Sequoia 15 on Apple M Series (ARMv8.6-A) M2 Pro
- macOS Sequoia 15 on Apple M Series (ARMv8.6-A) M2 Max
- macOS Sequoia 15 on Apple M Series (ARMv8.6-A) M2 Ultra
- macOS Sequoia 15 on Apple M Series (ARMv8.6-A) M3
- macOS Sequoia 15 on Apple M Series (ARMv8.6-A) M3 Pro
- macOS Sequoia 15 on Apple M Series (ARMv8.6-A) M3 Max
- macOS Sequoia 15 on Apple M Series (ARMv9.2-A) M4
- macOS Sequoia 15 on Apple M Series (ARMv9.2-A) M4 Pro
- macOS Sequoia 15 on Apple M Series (ARMv9.2-A) M4 Max

The SE cryptographic implementation executes on the following operational environments:

- Apple M Series M1
- Apple M Series (ARMv8.5-A) M1 Pro
- Apple M Series (ARMv8.5-A) M1 Max
- Apple M Series (ARMv8.5-A) M1 Ultra
- Apple M Series (ARMv8.6-A) M2
- Apple M Series (ARMv8.6-A) M2 Pro
- Apple M Series (ARMv8.6-A) M2 Max
- Apple M Series (ARMv8.6-A) M2 Ultra
- Apple M Series (ARMv8.6-A) M3
- Apple M Series (ARMv8.6-A) M3 Pro
- Apple M Series (ARMv8.6-A) M3 Max
- Apple M Series (ARMv9.2-A) M4
- Apple M Series (ARMv9.2-A) M4 Pro
- Apple M Series (ARMv9.2-A) M4 Max

The BC cryptographic implementation executes on the following operational environments:

- Crypto Hardware Module aes_core_gcm.vhd
- Crypto Hardware Module aes_core_gcm_simult_enc_mic.vhd
- Mentor Questa Sim-64 2021.2_1

The tables below show the cryptographic services used by the TOE and provided by the cryptographic modules, describing the algorithms, their supported key sizes, applicable standard and purpose. The tables also include the certificates obtained from the Cryptographic Algorithm Validation Program (CAVP) in the evaluated configuration for each of the cryptographic algorithms.

Table 13: Mapping of SFRs to CAVP certificates (USR cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_CKM.1	RSA	Modulus: 3072, 4096 bits	[FIPS186-5]	A6512
	ECC	Curves: P-256, P-384, P-521	[FIPS186-5]	A6512
FCS_CKM.2	RSA Key Establishment	Modulus: 3072, 4096 bits	[RFC8017]	CCTL Tested
	KAS-ECC-SSC	Curves: P-256, P-384, P-521	[SP800-56Ar3]	A6510
FCS_COP.1/ENCRYPT	AES-KW	256 bits encrypt, decrypt	[SP800-38F]	A6508
	AES-GCM	256 bits encrypt, decrypt	[SP800-38D]	A6511
FCS_COP.1/HASH	SHA2-256, SHA2-384, SHA2-512	Byte-oriented mode	[FIPS180-4]	A6513
FCS_COP.1/SIGN	RSA SigGen	Modulus: 3072, 4096 bits Hash: SHA2-256, SHA2-384, SHA2-512 Padding: PKCS#1 v1.5 and PSS	[FIPS186-5]	A6512
	RSA SigVer	Modulus: 3072, 4096 bits Hash: SHA2-256, SHA2-384, SHA2-512 Padding: PKCS#1 v1.5 and PSS	[FIPS186-5]	A6512
	ECDSA SigGen	Curves: P-384, P-521 Hash: SHA2-256, SHA2-384, SHA2-512	[FIPS186-5]	A6512
	ECDSA SigVer	Curves: P-384, P-521 Hash: SHA2-256, SHA2-384, SHA2-512	[FIPS186-5]	A6512
FCS_COP.1/KEYHMAC	HMAC-SHA-256, HMAC-SHA-384	Byte-oriented mode	[FIPS198-1]	A6513
FCS_RBG_EXT.1	CTR_DRBG	AES-256	[SP800-90Ar1]	A6511

Table 14: Mapping of SFRs to CAVP certificates (KRN cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1/HASH	SHA2-384	Byte-oriented mode	[FIPS180-4]	A6408

FCS_COP.1/SIGN	RSA SigVer	Modulus: 4096 bits Hash: SHA2-384 Padding: PKCS#1 v1.5	[FIPS186-5]	A6407
FCS_RBG_EXT.1	CTR_DRBG	AES-256	[SP800-90Ar1]	A6406

Table 15: Mapping of SFRs to CAVP certificates (SKS cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1/HASH	SHA2-384	Byte-oriented mode	[FIPS180-4]	A6560
FCS_COP.1/SIGN	RSA SigVer	Modulus: 4096 bits Hash: SHA2-384 Padding: PKCS#1 v1.5	[FIPS186-5]	A6560

Table 16: Mapping of SFRs to CAVP certificates (SE cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_RBG_EXT.1	CTR_DRBG	AES-256	[SP800-90Ar1]	A1362, A3490, A6548

Table 17: Mapping of SFRs to CAVP certificates (BC cryptographic implementation)

SFR	Algorithm	Capabilities	Standard	CAVP cert.
FCS_COP.1/ENCRYPT	AES-CTR	256 bits encrypt, decrypt	[SP800-38A]	AES 5926, AES 5927
	AES-CCM	128 bits encrypt, decrypt	[SP800-38C]	AES 5926, AES 5927, A1932
	AES-CCM	256 bits encrypt, decrypt	[SP800-38C]	AES 5952, AES 5953, A1932
	AES-GCM	256 bits encrypt, decrypt	[SP800-38D]	AES 5926, AES 5927, A1932

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

The TOE supports generation of 3072-bit, and 4096-bit RSA keys for use in TLS sessions with client authentication.

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.

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. It is used for the generation of AES keys of 256 bits. The AES-CCMP-256 and AES-GCMP-256 cryptographic algorithms are used to secure the Wi-Fi data traffic.

The developer employs internal testing and independent external testing to ensure that the TOE implementation conforms to cryptographic standards. Each Wi-Fi chip integrated into the TOE hardware platforms is covered by a Wi-Fi Alliance certificate. Additionally, the developer performs extensive internal testing (approximately 4800 test cases) covering a wide range of Wi-Fi features, from general interoperability to cryptographic support. The cryptographic testing includes verification of algorithms (AES-CCMP-256, AES-GCMP-256) and security protocols (WPA2 and WPA3) against the Wi-Fi standards.

7.1.2.2 FCS_CKM.2 - Cryptographic Key Establishment

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.

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).

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

The TOE performs Group Temporal Key (GTK) unwrapping in accordance with the following key distribution method: AES Key Wrap (as defined in RFC 3394) in an Extensible Authentication Protocol over LAN (EAPOL) key frame (as defined in IEEE 802.11-2012).

7.1.2.4 FCS_CKM_EXT.4 - Cryptographic Key Destruction

The TOE includes the Keychain Access app that allows users the ability to add, remove, and manage certificates, shared secrets, and private keys. Please see Section 7.1.2.11 for the details of Keychain Access. Each keychain item is protected with an individual Data Encryption Key (DEK). The metadata of all keychain items is collectively encrypted with another DEK (the metadata key).

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 as described in Section 7.1.4.8.

All 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).

TLS private keys and certificates are stored encrypted in the keychain. 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 are zeroized when no longer needed.

Bluetooth SSP private keys and DEKs are not stored persistently. Rather, the Bluetooth “Link Key” is stored encrypted in the keychain. The Link Key is used to derive the DEK.

The TOE does not store the Pairwise Master Key (PMK), Pairwise Transient Key (PTK), or Group Transient Key (GTK) in non-volatile storage.

Wrapped keys held in non-volatile storage (KEK and DEK) are deleted by invoking an interface provided by the underlying platform that destroys the abstraction that represents the key. Keys held in volatile memory are destroyed via single overwrite consisting of zeroes.

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

7.1.2.5 FCS_CKM_EXT.8 - Bluetooth Key Generation

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

7.1.2.6 FCS_COP.1/ENCRYPT - Cryptographic Operation - Encryption/Decryption

The TOE supports AES encryption using 128-bit and 256-bit keys. The 128-bit key is only used with AES-CCM for Bluetooth functions.

7.1.2.7 FCS_COP.1/HASH - Cryptographic Operation - Hashing

Hashing algorithms are used for signature generation and verification, and HMAC. Table 13, Table 14, and Table 15 identify the hash functions used by the signature algorithms.

7.1.2.8 FCS_COP.1/SIGN - Cryptographic Operation - Signing

The TOE performs signature generation and verification as described in Section 6.1.2.9.

7.1.2.9 FCS_COP.1/KEYHMAC - Cryptographic Operation - Keyed-Hash Message Authentication

The TOE performs keyed-hash message authentication as described in Section 6.1.2.10.

7.1.2.10 FCS_RBG_EXT.1 - Random Bit Generation

The TOE performs random bit generation as described in Section 6.1.2.11 and the proprietary Entropy Assessment Report.

7.1.2.11 FCS_STO_EXT.1 Storage of Sensitive Data

The TOE offers an encrypted database, called keychain, to securely store sensitive data. Users can access the keychain by opening the Keychain Access app in /Applications/Utilities/. An initial default keychain is created for each user, though users can create other keychains for specific purposes. 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 TOE stores the following sensitive data in the keychain:

- Trusted certificate authorities for establishing TLS sessions.
- Private keys for establishing TLS sessions.
- Bluetooth Link Keys to manage the pairing and secure communication with Bluetooth devices.
- Wi-Fi passwords to authenticate the TOE itself when connecting to wireless access points.

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 while the secret value (kSecValueData) is encrypted with the secret key. Both keys are wrapped by KEKs stored in the Secure Enclave. The metadata key is cached in the application processor to allow fast queries of the keychain.

7.1.2.12 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.4 TLS Client Support for Renegotiation, FCS_TLSC_EXT.5 - TLS Client Support for Supported Groups Extension

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

The TOE establishes the reference identifier by parsing the Domain Name System (DNS) Name for the configured TLS server. The reference identifier is matched against the Subject Alternative Name (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 Universal Resource Identifier (URI) reference identifiers, DNS Service (SRV) reference identifiers, or IP address reference identifiers.

The TOE supports certificate pinning using the TLS framework. However, existing TLS clients in the TOE, such as the Safari browser, do not support certificate pinning. WLAN also does not support certificate pinning.

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

The TOE supports mutual authentication using X.509v3 certificates.

The TOE supports the Supported Groups Extension by default with the following supported groups: secp384r1 and secp521r1.

7.1.2.13 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)

For the Wi-Fi EAP-TLS client, the TOE implements TLS 1.2 (RFC 5246) 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

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, SRV reference identifiers, or IP address reference identifiers.

The TOE supports the Supported Groups Extension by default with the following supported groups: secp384r1 and secp521r1.

7.1.2.14 FCS_WPA_EXT.1 - Supported WPA Versions

The TOE supports Wi-Fi Protected Access as described in Section 6.1.2.20.

7.1.3 User Data Protection (FDP)

The TOE includes the following file system security schemes providing access control to user data: sandbox entitlements, access control lists (ACLs), Berkeley Software Distribution (BSD) file flags, and standard Unix permissions.

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

- If the app's sandbox forbids the requested access, the request is denied.
- If an ACL exists on the file, it is evaluated to determine access rights.
- If a BSD file flag prohibits the operation, the operation is denied.
- Finally, the Unix permissions are evaluated to determine access rights:
  - If the user ID matches the owner of the file, the "owner" permissions are used.
  - Otherwise, if the group ID matches the group for the file, the "group" permissions are used.
  - Otherwise, the "other" permissions are used.

Sandbox Entitlements

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.

ACLs

The TOE supports ACLs, which provide more fine-grained access control to files and directories than the Unix permissions. For example, ACLs allow the system administrator to specify that a specific user can append new contents to a file but cannot modify the existing contents of the file.

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. The TOE supports ACL inheritance such that the files and subdirectories created within a directory can inherit the ACL of the parent directory.

BSD File Flags

As a BSD-derived system, the TOE supports the BSD file flags, which provide additional security control over files and directories. For example, the file owner can use the BSD file flag "uchange" to specify that a file cannot be deleted. The BSD file flags override the Unix permissions.

Standard Unix Permissions

The UNIX permission settings for a file or directory are a 9-bit string that is often represented as a three-digit octal value. The “owner”, “group”, and “other” bit sets contain three bits respectively: read, write, execute (rwx for short), indicating separate read, write, and execute permissions for the "owner", "group", and "other".

The "owner" permissions apply to the user who owns the file or directory. The "group" permissions apply to all users who belong to the group associated with the file or directory. The "other" permissions apply to all remaining users.

7.1.4 Identification and Authentication (FIA)

7.1.4.1 FIA_AFL.1 - Authentication Failure Handling

The TOE handles authentication failures as described in Section 6.1.4.1.

7.1.4.2 FIA_BLT_EXT.1 - Bluetooth User Authorization

Explicit user authorization is required for both pairing and removing a Bluetooth device from the TOE's device list.

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. Finally, the user could also affirm the remote device name depending on the remote Bluetooth device.

7.1.4.3 FIA_BLT_EXT.2 - Bluetooth Mutual Authentication

The TOE's Bluetooth device driver prevents data transfer via Bluetooth until pairing has fully completed and the devices have mutually authenticated. The TOE requires manual authorization from the user during the pairing process, through showing the user a system dialog (e.g., PIN request, confirmation). The TOE provides no APIs for programmatically bypassing the system pairing dialogs.

The TOE supports the Object Exchange (OBEX) protocol, which runs over the Radio Frequency Communication (RFCOMM) protocol, which in turn runs on top of the Logical Link Control and Adaptation Layer Protocol (L2CAP). The TOE also supports the Audio/Video Control Transport Protocol (AVCTP), the Audio/Video Distribution Transport Protocol (AVDTP), the Service Discovery Protocol (SDP), and the Attribute Protocol (ATT).

7.1.4.4 FIA_BLT_EXT.3 - Rejection of Duplicate Bluetooth Connections

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.

The process for starting a Bluetooth connection covers the handling of duplicates in the Bluetooth layers of the TOE devices. Depending on the Bluetooth chip, it could either be the macOS Bluetooth controller or software orchestrating the process with support from the Link Manager Protocol (LMP) or the Host Controller Interface (HCI) layer.

The process is as follows:

1. Connection List: A connection list is maintained to track all active Bluetooth connections by device identifier (BD_ADDR).
2. Pre-Connection Validation: When an incoming Bluetooth connection request arrives at the LMP or HCI layer, the device invokes a callback that queries the connection list before the connection is accepted.
3. Duplicate Detection: The layer managing the Bluetooth connections (LMP or HCI) performs a lookup in the connection list using the device's BD_ADDR.
4. Rejection: If the lookup returns true (i.e., the device is already connected), that layer rejects the connection request before completing the connection establishment.

7.1.4.5 FIA_BLT_EXT.4 - Secure Simple Pairing

Devices that want to pair with the TOE via Bluetooth can use Secure Simple Pairing (SSP). 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.

7.1.4.6 FIA_BLT_EXT.6 - Trusted Bluetooth Device User Authorization, FIA_BLT_EXT.7 - Untrusted Bluetooth Device User Authorization

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

- Hands-Free Profile (HFP 1.8)
- Advanced Audio Distribution Profile (A2DP 1.3)
- Audio/Video Remote Control Profile (AVRCP 1.5, AVRCP 1.6)
- Human Interface Device Profile (HID)
- Generic Attribute Profile (GATT)
- Serial Port Profile (SPP 1.2)

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 the following two cases:

- 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 through the pairing process.

7.1.4.7 FIA_PAE_EXT.1 - Port Access Entity Authentication

The TOE supports port-based network access control as described in Section 6.1.4.8.

7.1.4.8 FIA_UAU.5 - Multiple Authentication Mechanisms

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

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's key bags using AES-KW. If the output key from the PBKDF2 function successfully unwraps the user's key bags, the user is authenticated and granted access; otherwise, the user is denied access.

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. 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 encryption key is then used to unwrap the user's key bags.

7.1.4.9 FIA_X509_EXT.1 X.509 - Certificate Validation, FIA_X509_EXT.1/WLAN X.509 - Certificate Validation

When an X.509 certificate is presented for authentication, the TOE verifies the certificate path as described in Sections 6.1.4.10 and 6.1.4.11.

X.509 certificates are validated when imported into the keychain, during session establishment with a TLS or EAP-TLS server, and prior to presenting a certificate to the server during TLS or EAP-TLS mutual authentications.

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

The TOE uses X.509v3 certificates for performing mutual authentication for TLS connections. The TOE also uses X.509v3 certificates to support authentication for 802.1X EAP-TLS exchanges.

The TOE stores digital certificates in the keychain. The keychain contains a set of pre-installed Certification Authority (CA) certificates, and additional trusted CA certificates or client certificates can be imported by users or applications.

Each certificate stored in the keychain is associated with a trust policy. A trust policy specifies whether a certificate is trusted and for which intended purposes. A certificate may be valid for certain uses (e.g., Extensible Authentication) but not for others (e.g., Code Signing). By configuring the trust policy, users can control whether a certificate is considered trustworthy and define the conditions under which it may be used.

7.1.4.11 FIA_X509_EXT.6 - X.509 Certificate Storage and Management

The TOE ensures that all certificates on the system are securely stored and protected from unauthorized deletion or modification. The certificates are stored in an encrypted database, called keychain.

Keychain items are encrypted with AES-256-GCM, which provides both confidentiality and integrity protection. The user is required to enter the correct Administrator password before modifying the system-level keychains, such as loading certificates into the keychain.

7.1.5 Security Management (FMT)

Functions requiring Administrator access require the user to enter the correct Administrator password before allowing the user to modify the function.

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.

The TOE supports both Bluetooth BR/EDR and LE, can make use of Secure Simple Pairing for security, and supports the Bluetooth profiles listed in Section 7.1.4.6.

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).

7.1.6 Protection of the TSF (FPT)

7.1.6.1 FPT_ACF_EXT.1 - Access Controls

System Integrity Protection restricts the root user account and limits the actions that the root user can perform on protected parts of the Mac operating system. System Integrity Protection protects the following parts of the system from unauthorized modification:

a. Kernel drivers/modules:
    i. /System/Library/Extensions/
b. Shared libraries:
    i. /usr/lib
    ii. /Library/Frameworks
    iii. /System/Library/Frameworks
    iv. /System/Library/PrivateFrameworks
c. System executables:
    i. /usr/bin
d. Applications
    i. /Applications - Apps that are installed with the TOE but updated independently
    ii. /System/Applications
    iii. /System/Library/CoreServices

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. Apps downloaded from the App Store already work with System Integrity Protection.

Standard file permissions 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. /private/etc
    ii. /Library/Preferences
c. TSF data:
    i. /var
    ii. /var/folders/ - user-specific TSF data. A user has permission to modify their own data
d. System-wide credentials repositories:
    i. /Library/Keychains/, encrypted using /var/db/SystemKey
e. Applications:
    i. /Applications - Apps installed by a traditional installer and Apps copied into /Applications

7.1.6.2 FPT_ASLR_EXT.1 - Address Space Layout Randomization (ASLR)

The TOE implements address space layout randomization as described in Section 6.1.6.2.

7.1.6.3 FPT_SBOP_EXT.1 - Stack Buffer Overflow Protection

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

- ASLR to randomize locations on the stack, preventing attackers from jumping to specific data that has been written to the stack.
- Stack canaries to detect if the stack has been overwritten when returning from a function.

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:

- Type 1: The compiler can optimize away stack usage (which macOS heavily relies on for performance reasons).
- 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.
- 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).
- 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.
- Type 5: Some binaries do not access the stack in any kind of vulnerable way.

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

The boot process on Apple silicon devices is as follows:

1) The processor loads the Boot ROM.
2) The Boot ROM validates the Low-Level Bootloader (LLB) signature using the Apple Root CA public key.
3) LLB validates system-paired firmware signatures.
4) LLB validates iBoot stage 2 signature.
5) iBoot stage 2 validates the macOS-paired firmware, Boot Kernel Collection, Auxiliary Kernel Collection (if applicable), system trust cache, and signed system volume signatures.
6) The TOE (macOS) begins execution.

The boot process for sepOS is as follows:

1) iBoot assigns a dedicated region of memory to the Secure Enclave.
2) The processor sends the sepOS image to the Secure Enclave Processor, which is executing the Secure Enclave Boot ROM.
3) The Secure Enclave Processor checks the digital signature of the sepOS image.
4) If the signature is deemed valid, sepOS begins execution.

The Boot ROM 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.

The TOE performs software integrity testing on the runtime image of each implemented cryptographic 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.

The WLAN client relies on cryptographic functionality implemented by the “BC” chip identified in Section 7.1.2 which is embedded in the underlying platform. The TOE verifies that the chip has successfully completed its own 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.

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.

7.1.6.5 FPT_TUD_EXT.1 - Trusted Update, FPT_TUD_EXT.2 - Trusted Update for Application Software

The TOE allows the user to check for and install OS updates using the Software Update setting pane.

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

7.1.7 TOE Access (FTA)

7.1.7.1 FTA_TAB.1 - Default TOE Access Banners

The TOE implements access banners as described in Section 6.1.7.1.

7.1.7.2 FTA_WSE_EXT.1 - Wireless Network Access

Administrators can restrict the wireless networks to which the TOE device connects through specifying a list of Known Networks (identified by SSID) and configuring the TOE to require an administrator password to switch to a different Wi-Fi network.

7.1.8 Trusted Path (FTP)

7.1.8.1 FTP_BLT_EXT.1 - Bluetooth Encryption

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

Devices that want to pair with the TOE via Bluetooth are required by the TOE to use Secure Simple Pairing, which uses ECDH key exchange and AES for data encryption.

7.1.8.2 FTP_BLT_EXT.2 - Persistence of Bluetooth Encryption

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

7.1.8.3 FTP_BLT_EXT.3/BR - Bluetooth Encryption Parameters (BR/EDR), FTP_BLT_EXT.3/LE - Bluetooth Encryption Parameters (LE)

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.

7.1.8.4 FTP_ITC_EXT.1 - Trusted Channel Communication, FTP_ITC.1/WLAN - Trusted Channel Communication (Wireless LAN)

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.

7.1.8.5 FTP_TRP.1 - Trusted Path

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.

7.2 ALC_TSU_EXT.1 - Timely Security Updates

Security issues reported to Apple are screened and categorized by severity level, and a report is established. Most reports are resolved within 90 days. More severe issues are prioritized for earlier resolution. Security updates for validated vulnerabilities will be scheduled on target releases based on severity.

Apple does not disclose, discuss, or confirm security issues until an investigation has occurred and patches or releases are generally available. Once an issue has been confirmed and a security update has been made available, references containing technical details are made available and Common Vulnerabilities and Exposures (CVEs) are released. Security updates are made available via the Apple Update Server the same day that security issues are announced. Therefore, there are 0 days between public disclosure of a vulnerability and public availability of the TOE security update patching this vulnerability.

Apple publishes security notifications and announcements on the web at "Apple security releases" page (https://support.apple.com/100100). That page provides the information about security updates including the specific CVEs addressed by each update. The security notifications and announcements are also distributed through the Security-announce mailing list (https://lists.apple.com/mailman3/lists/security-announce.lists.apple.com/).

The instructions of reporting security issues are available on the web at "Report a security or privacy vulnerability" page (https://support.apple.com/en-us/102549). Users can report security issues related to the TOE on the web at Apple Security Research (https://security.apple.com/). Alternatively, they can send email to "product-security@apple.com", using Apple Product Security PGP key (https://support.apple.com/en-us/102148) to encrypt the email.

A. Devices Covered by this Evaluation

The evaluated configuration includes the following Apple devices:

Table 18: Hardware Platforms

Marketing Name	Model	Model Identifier	Processor (Micro Architecture)	Security Chip	BT Version (BT/Wi-Fi Chip)
2025					
MacBook Air (15-inch, M4, 2025)	A3241	Mac16,13	M4 (ARMv9.2-A)	SEP v2.0	5.3 (4388)
MacBook Air (13-inch, M4, 2025)	A3240	Mac16,12	M4 (ARMv9.2-A)	SEP v2.0	5.3 (4388)
Mac Studio (2025)	A3143	Mac16,9	M4 Max (ARMv9.2-A)	SEP v2.0	5.3 (4388)
2024					
MacBook Pro (14-inch, 2024)	A3401	Mac16,8	M4 Pro (ARMv9.2-A)	SEP v2.0	5.3 (4388)
	A3185	Mac16,6	M4 Max (ARMv9.2-A)	SEP v2.0	5.3 (4388)
	A3112	Mac16,1	M4 (ARMv9.2-A)	SEP v2.0	5.3 (4388)
MacBook Pro (16-inch, 2024)	A3403	Mac16,7	M4 Pro (ARMv9.2-A)	SEP v2.0	5.3 (4388)
	A3186	Mac16,5	M4 Max (ARMv9.2-A)	SEP v2.0	5.3 (4388)
iMac (24-inch, 2024, Four ports)	A3137	Mac16,3	M4 (ARMv9.2-A)	SEP v2.0	5.3 (4388)
	A3247	Mac16,2	M4 (ARMv9.2-A)	SEP v2.0	5.3 (4388)
Mac mini (2024)	A3239	Mac16,11	M4 Pro (ARMv9.2-A)	SEP v2.0	5.3 (4388)
	A3238	Mac16,10	M4 (ARMv9.2-A)	SEP v2.0	5.3 (4388)
MacBook Air (15-inch, M3, 2024)	A3114	Mac15,13	M3 (ARMv8.6-A)	SEP v2.0	5.3 (4388)
MacBook Air (13-inch, M3, 2024)	A3113	Mac15,12	M3 (ARMv8.6-A)	SEP v2.0	5.3 (4388)
2023					
MacBook Pro (14-inch, Nov 2023)	A2992	Mac15,10	M3 Max (ARMv8.6-A)	SEP v2.0	5.3 (4388)
		Mac15,8	M3 Max (ARMv8.6-A)	SEP v2.0	5.3 (4388)
		Mac15,6	M3 Pro (ARMv8.6-A)	SEP v2.0	5.3 (4388)
	A2918	Mac15,3	M3 (ARMv8.6-A)	SEP v2.0	5.3 (4388)
MacBook Pro (16-inch, Nov 2023)	A2991	Mac15,11	M3 Max (ARMv8.6-A)	SEP v2.0	5.3 (4388)
		Mac15,9	M3 Max (ARMv8.6-A)	SEP v2.0	5.3 (4388)

		Mac15,7	M3 Pro (ARMv8.6-A)	SEP v2.0	5.3 (4388)
iMac (24-inch, 2023)	A2873	Mac15,5	M3 (ARMv8.6-A)	SEP v2.0	5.3 (4388)
	A2874	Mac15,4	M3 (ARMv8.6-A)	SEP v2.0	5.3 (4388)
Mac Pro (-/Rack 2023)	A2786	Mac14,8	M2 Ultra (ARMv8.6-A)	SEP v2.0	5.3 (4388)
MacBook Air (15-inch, 2023)	A2941	Mac14,15	M2 (ARMv8.6-A)	SEP v2.0	5.3 (4387)
Mac Studio (2023)	A2901	Mac14,14	M2 Ultra (ARMv8.6-A)	SEP v2.0	5.3 (4388)
		Mac14,13	M2 Max (ARMv8.6-A)	SEP v2.0	5.3 (4388)
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)	SEP v2.0	5.3 (4388)
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)	SEP v2.0	5.3 (4388)
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)	A2681	Mac14,2	M2 (ARMv8.6-A)	SEP v2.0	5.3 (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)	SEP v2.0	5.0 (4387)
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)	SEP v2.0	5.0 (4387)
MacBook Pro (14-inch, 2021)	A2442	MacBookPro18,4	M1 Max (ARMv8.5-A)	SEP v2.0	5.0 (4387)
		MacBookPro18,3	M1 Pro (ARMv8.5-A)	SEP v2.0	5.0 (4387)
iMac (24-inch, M1, 2021)	A2438	iMac21,1	M1 (ARMv8.5-A)	SEP v2.0	5.0 (4378)
	A2439	iMac21,2	M1 (ARMv8.5-A)	SEP v2.0	5.0 (4378)
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)

B. Abbreviations and Terminology

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

BSM

Basic Security Module

CA

Certificate Authority

CAVP

Cryptographic Algorithm Validation Program

CC

Common Criteria

CCM

Counter with CBC-MAC

CCMP

CCM Mode Protocol

CEM

Common Evaluation Methodology

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

EAP

Extensible Authentication Protocol

EAPOL

Extensible Authentication Protocol over LAN

ECC

Elliptic Curve Cryptography

ECDH

Elliptic Curve Diffie-Hellman

ECDHE

ECDH Ephemeral

EKU

extendedKeyUsage

EST

Enrollment over Secure Transport

FIPS

Federal Information Processing Standards

GCM

Galois/Counter Mode

GCMP

Galois Counter Mode Protocol

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

LMP

Link Manager Protocol

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

PIN

Personal Identification Number

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

SL

Security Level (FIPS 140-3)

SRV

DNS Service

SSID

Service Set Identifier

SSP

Secure Simple Pairing

ST

Security Target

TLS

Transport Layer Security

TOE

Target of Evaluation

TRNG

True Random Number Generator

TSF

TOE Security Function

TSFI

TSF Interface

TSS

TOE Summary Specification

UID

User Identifier

URI

Universal Resource Identifier

UUID

Universally Unique Identifier

WLAN

Wireless Local Area Network

WLANC

Wireless Local Area Network Client

WPA

Wi-Fi Protected Access

XNU

X is Not Unix