

**Apple Inc.**



**Apple macOS 13 Ventura  
Security Target**

<b>Version:</b>	1.1
<b>Status:</b>	Final
<b>Last Update:</b>	2024-01-12
<b>Validation Body:</b>	NIAP
<b>Validation ID:</b>	VID11347
<b>Classification:</b>	Public

## 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.

## 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 and distributed only in its original entirety without revision.

## Revision History

Version	Date	Author(s)	Changes to Previous Revision
1.0	2023-11-22	Alejandro Masino	First published version.
1.1	2024-01-12	Alejandro Masino	Address NIAP's ECR comments.

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>8</b>
1.1	Security Target Identification	8
1.2	TOE Identification	8
1.3	TOE Type	8
1.4	TOE Overview	8
1.5	TOE Description	8
1.5.1	Architecture	8
1.5.2	Physical boundary	9
1.5.3	TOE security functionality	9
1.5.3.1	Security audit	9
1.5.3.2	Cryptographic support	9
1.5.3.3	User data protection	10
1.5.3.4	Identification and authentication	10
1.5.3.5	Security management	10
1.5.3.6	Protection of the TSF	10
1.5.3.7	TOE access	10
1.5.3.8	Trusted path/channels	10
1.5.4	TOE operational environment	10
<b>2</b>	<b>CC Conformance Claim</b>	<b>12</b>
2.1	Protection Profile Tailoring and Additions	12
2.1.1	Protection Profile for General Purpose Operating Systems ([OSPPv4.2.1])	12
2.1.2	PP-Module for Bluetooth ([BT])	13
<b>3</b>	<b>Security Problem Definition</b>	<b>14</b>
3.1	Threat Environment	14
3.1.1	Threats countered by the TOE	14
3.2	Assumptions	14
<b>4</b>	<b>Security Objectives</b>	<b>15</b>
4.1	Objectives for the TOE	15
4.2	Objectives for the Operational Environment	15
4.3	Security Objectives Rationale	16
4.3.1	Coverage	16
4.3.2	Sufficiency	16
<b>5</b>	<b>Extended Components Definition</b>	<b>18</b>
<b>6</b>	<b>Security Requirements</b>	<b>19</b>
6.1	TOE Security Functional Requirements	19
6.1.1	Security audit (FAU)	21
6.1.1.1	FAU_GEN.1 Audit Data Generation (Refined)	21
6.1.1.2	FAU_GEN.1/BT Audit Data Generation (Bluetooth)	21
6.1.2	Cryptographic support (FCS)	23
6.1.2.1	FCS_CKM.1 Cryptographic Key Generation (Refined)	23
6.1.2.2	FCS_CKM.2 Cryptographic Key Establishment (Refined)	23
6.1.2.3	FCS_CKM_EXT.4 Cryptographic Key Destruction	23

6.1.2.4	FCS_CKM_EXT.8 Bluetooth Key Generation	24
6.1.2.5	FCS_COP.1(1) Cryptographic Operation - Encryption/Decryption (Refined)	24
6.1.2.6	FCS_COP.1(2) Cryptographic Operation - Hashing (Refined)	24
6.1.2.7	FCS_COP.1(3) Cryptographic Operation - Signing (Refined)	25
6.1.2.8	FCS_COP.1(4) Cryptographic Operation - Keyed-Hash Message Authentication (Refined)	25
6.1.2.9	FCS_RBG_EXT.1 Random Bit Generation	25
6.1.2.10	FCS_STO_EXT.1 Storage of Sensitive Data	26
6.1.2.11	FCS_TLSC_EXT.1 TLS Client Protocol	26
6.1.2.12	FCS_TLSC_EXT.2 TLS Client Protocol	26
6.1.2.13	FCS_TLSC_EXT.4 TLS Client Protocol	26
6.1.3	User data protection (FDP)	27
6.1.3.1	FDP_ACF_EXT.1 Access Controls for Protecting User Data	27
6.1.4	Identification and authentication (FIA)	27
6.1.4.1	FIA_AFL.1 Authentication failure handling (Refined)	27
6.1.4.2	FIA_BLT_EXT.1 Bluetooth User Authorization	27
6.1.4.3	FIA_BLT_EXT.2 Bluetooth Mutual Authentication	27
6.1.4.4	FIA_BLT_EXT.3 Rejection of Duplicate Bluetooth Connections	28
6.1.4.5	FIA_BLT_EXT.4 Secure Simple Pairing	28
6.1.4.6	FIA_BLT_EXT.6 Trusted Bluetooth Device User Authorization	28
6.1.4.7	FIA_BLT_EXT.7 Untrusted Bluetooth Device User Authorization	28
6.1.4.8	FIA_UAU.5 Multiple Authentication Mechanisms (Refined)	28
6.1.4.9	FIA_X509_EXT.1 X.509 Certificate Validation	29
6.1.4.10	FIA_X509_EXT.2 X.509 Certificate Authentication	30
6.1.5	Security management (FMT)	30
6.1.5.1	FMT_MOF_EXT.1 Management of security functions behavior	30
6.1.5.2	FMT_MOF_EXT.1/BT Management of Security Functions Behavior	30
6.1.5.3	FMT_SMF_EXT.1 Specification of Management Functions	30
6.1.5.4	FMT_SMF_EXT.1/BT Specification of Management Functions	31
6.1.6	Protection of the TSF (FPT)	32
6.1.6.1	FPT_ACF_EXT.1 Access controls	32
6.1.6.2	FPT_AS LR_EXT.1 Address Space Layout Randomization	32
6.1.6.3	FPT_SBOP_EXT.1 Stack Buffer Overflow Protection	32
6.1.6.4	FPT_TST_EXT.1 Boot Integrity	33
6.1.6.5	FPT_TUD_EXT.1 Trusted Update	33
6.1.6.6	FPT_TUD_EXT.2 Trusted Update for Application Software	33
6.1.6.7	FPT_W^X_EXT.1 Write XOR Execute Memory Pages	34
6.1.7	TOE access (FTA)	34
6.1.7.1	FTA_TAB.1 Default TOE access banners	34
6.1.8	Trusted path/channels (FTP)	34
6.1.8.1	FTP_BLT_EXT.1 Bluetooth Encryption	34
6.1.8.2	FTP_BLT_EXT.2 Persistence of Bluetooth Encryption	34
6.1.8.3	FTP_BLT_EXT.3/BR Bluetooth Encryption Parameters (BR/EDR)	34
6.1.8.4	FTP_BLT_EXT.3/LE Bluetooth Encryption Parameters (LE)	35
6.1.8.5	FTP_ITC_EXT.1 Trusted channel communication	35

6.1.8.6	FTP_TRP.1 Trusted Path .....	35
6.2	Security Functional Requirements Rationale .....	36
6.2.1	Coverage .....	36
6.2.2	Sufficiency .....	37
6.3	Security Assurance Requirements .....	39
6.3.1	ALC Life-cycle support .....	40
6.3.1.1	ALC_TSU_EXT.1 Timely Security Updates .....	40
6.4	Security Assurance Requirements Rationale .....	40
<b>7</b>	<b>TOE Summary Specification .....</b>	<b>41</b>
7.1	TSS Security Assurance Evaluation Activity .....	41
7.1.1	Timely security updates (ALC_TSU_EXT.1) .....	41
7.2	TOE Security Functionality .....	41
7.2.1	Audit .....	41
7.2.1.1	FAU_GEN.1 Audit data generation .....	41
7.2.1.2	FAU_GEN.1/BT Audit data generation (Bluetooth) .....	42
7.2.2	Cryptography .....	42
7.2.2.1	FCS_CKM.1 Cryptographic key generation .....	43
7.2.2.2	FCS_CKM.2 Cryptographic key establishment .....	44
7.2.2.3	FCS_CKM_EXT.4 Cryptographic key destruction .....	44
7.2.2.4	FCS_CKM_EXT.8 Bluetooth key generation .....	45
7.2.2.5	FCS_COP.1(1) Cryptographic operation - encryption/decryption .....	45
7.2.2.6	FCS_COP.1(2) Cryptographic operation - hashing .....	45
7.2.2.7	FCS_COP.1(3) Cryptographic operation - signing .....	45
7.2.2.8	FCS_COP.1(4) Cryptographic operation - keyed-hash message authentication .....	45
7.2.2.9	FCS_RBG_EXT.1 Random bit generation .....	46
7.2.2.10	FCS_STO_EXT.1 Storage of sensitive data .....	46
7.2.2.11	FCS_TLSC_EXT.1 TLS client protocol .....	46
7.2.2.12	FCS_TLSC_EXT.2 TLS client protocol .....	47
7.2.2.13	FCS_TLSC_EXT.4 TLS client protocol .....	47
7.2.3	User data protection .....	47
7.2.3.1	FDP_ACF_EXT.1 Access controls for protecting user data .....	47
7.2.4	Identification and authentication .....	49
7.2.4.1	FIA_AFL.1 Authentication failure handling .....	49
7.2.4.2	FIA_BLT_EXT.1 Bluetooth user authorization .....	49
7.2.4.3	FIA_BLT_EXT.2 Bluetooth mutual authentication .....	50
7.2.4.4	FIA_BLT_EXT.3 Rejection of duplicate Bluetooth connections .....	50
7.2.4.5	FIA_BLT_EXT.4 Secure Simple Pairing .....	50
7.2.4.6	FIA_BLT_EXT.6 Trusted Bluetooth device user authorization .....	50
7.2.4.7	FIA_BLT_EXT.7 Untrusted Bluetooth device user authorization .....	51
7.2.4.8	FIA_UAU.5 Multiple authentication mechanisms .....	51
7.2.4.9	FIA_X509_EXT.1 X.509 Certificate validation .....	51
7.2.4.10	FIA_X509_EXT.2 X.509 Certificate authentication .....	52
7.2.5	Security management .....	52
7.2.5.1	FMT_MOF_EXT.1 Management of security functions behavior .....	52

7.2.5.2	FMT_MOF_EXT.1/BT Management of security functions behavior	52
7.2.5.3	FMT_SMF_EXT.1 Specification of management functions	52
7.2.5.4	FMT_SMF_EXT.1/BT Specification of management functions	53
7.2.6	Protection of the TSF	54
7.2.6.1	FPT_ACF_EXT.1 Access controls	54
7.2.6.2	FPT_ASLR_EXT.1 Address space layout randomization (ASLR)	55
7.2.6.3	FPT_SBOP_EXT.1 Stack buffer overflow protection	55
7.2.6.4	FPT_TST_EXT.1 Boot integrity	55
7.2.6.5	FPT_TUD_EXT.1 Trusted update	56
7.2.6.6	FPT_TUD_EXT.2 Trusted update for application software	57
7.2.6.7	FPT_W^X_EXT.1 Write XOR execute memory pages	57
7.2.7	TOE access	57
7.2.7.1	FTA_TAB.1 Default TOE access banners	57
7.2.8	Trusted path/channels	57
7.2.8.1	FTP_BLT_EXT.1 Bluetooth encryption	57
7.2.8.2	FTP_BLT_EXT.2 Persistence of Bluetooth encryption	58
7.2.8.3	FTP_BLT_EXT.3/BR Bluetooth encryption parameters (BR/EDR)	58
7.2.8.4	FTP_BLT_EXT.3/LE Bluetooth encryption parameters (LE)	58
7.2.8.5	FTP_ITC_EXT.1 Trusted channel communication	58
7.2.8.6	FTP_TRP.1 Trusted path	58
<b>8</b>	<b>Abbreviations, Terminology, and References</b>	<b>59</b>
8.1	Abbreviations	59
8.2	Terminology	63
8.3	References	65
<b>A</b>	<b>Appendixes</b>	<b>67</b>
<b>A.1</b>	<b>Hardware Platforms Covered by this Evaluation</b>	<b>67</b>
<b>A.2</b>	<b>SFR to CAVP Mapping Table</b>	<b>70</b>

## List of Tables

Table 1: TOE operational environment .....	11
Table 2: NIAP TDs for OSPP .....	12
Table 3: NIAP TDs for BT .....	13
Table 4: Mapping of security objectives to threats and policies .....	16
Table 5: Mapping of security objectives for the Operational Environment to assumptions, threats and policies .....	16
Table 6: Sufficiency of objectives countering threats .....	16
Table 7: Sufficiency of objectives holding assumptions .....	17
Table 8: SFRs for the TOE .....	19
Table 9: Auditable events (Bluetooth) .....	22
Table 10: Management functions (OSPP) .....	30
Table 11: Management functions (Bluetooth) .....	31
Table 12: Mapping of security functional requirements to security objectives .....	36
Table 13: Security objectives for the TOE rationale .....	37
Table 14: SARs .....	39
Table 15: Cryptographic algorithm table .....	43
Table 16: Hardware platforms .....	67
Table 17: Cryptographic algorithm table .....	70
Table 18: Coverage of CAVP certificates for Apple silicon .....	73
Table 19: Coverage of CAVP certificates for Intel Processors .....	73
Table 20: Coverage of CAVP certificates for Apple T2 Security Chip .....	75
Table 21: Coverage of CAVP certificates for Broadcom Chip with Bluetooth .....	75

# 1 Introduction

## 1.1 Security Target Identification

Title:	Apple macOS 13 Ventura Security Target
Version:	1.1
Status:	Final
Date:	2024-01-12
Sponsor:	Apple Inc.
Developer:	Apple Inc.
Validation Body:	NIAP
Validation ID:	VID11347
Keywords:	macOS, operating system

## 1.2 TOE Identification

The TOE is Apple macOS 13 Ventura.

## 1.3 TOE Type

The TOE type is general purpose operating system.

## 1.4 TOE Overview

The Security Target (ST) serves as the basis for the Common Criteria (CC) evaluation and identifies the Target of Evaluation (TOE), the scope of the evaluation, and the assumptions made throughout. This document will also describe the intended operational environment of the TOE and the functional and assurance requirements that the TOE meets.

The TOE is the Apple macOS 13 Ventura general purpose operating system (GPOS). The TOE is tightly integrated with hardware and runs on Apple iMac, MacBook Air, MacBook Pro, Mac mini, Mac Pro, and Mac Studio computers. The macOS Ventura operating system is a Unix-based graphical operating system. The macOS core is a POSIX compliant operating system built on top of the XNU kernel with standard Unix facilities available from the command line interface. The TOE includes Bluetooth communication—both Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) and Low Energy (LE). A portion of the TOE's Bluetooth functionality is implemented in hardware (Broadcom BT chip).

The tested version of the TOE is:

- Apple macOS 13.2.1

## 1.5 TOE Description

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

### 1.5.1 Architecture

The TOE is the software running on the Macs listed in [Appendix A.1 "Hardware Platforms Covered by this Evaluation"](#). These Macs are organized into the following two groups:



- Apple silicon Macs
- "Intel with T2" Macs

The Apple silicon Macs group represents all systems listed in [Appendix A.1](#) that use an Apple silicon System on a Chip (SoC). The "Intel with T2" Macs group represents all systems listed in [Appendix A.1](#) that use an Intel processor with the Apple T2 Security Chip. These groups have implementation differences where indicated in this document.

The Macs in this evaluation contain the Apple Secure Enclave. The Secure Enclave contains the Secure Enclave Processor (SEP) and a true random number generator (TRNG). The SEP runs sepOS, which is included with macOS and is within the TOE boundary. The TRNG is mentioned here as a tie-in with the Entropy Assessment Report (EAR), which is a document required by the conformance claims defined in [Section 2](#).

On Apple silicon Macs, the Secure Enclave is located on the SoC along with the application processor. On "Intel with T2" Macs, the Secure Enclave is located on the T2 chip.

The executing TOE is divided into user space and kernel space. User space contains processes that each 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 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.

Both the Apple silicon and "Intel with T2" Macs include a Broadcom chip that implements part of the bluetooth functionality; the chip model depends on the hardware platform and are also listed in [Appendix A.1](#).

## 1.5.2 Physical boundary

The physical boundary of the TOE is the installation image. The installation image includes both macOS and sepOS. The hardware platforms covered by this evaluation are listed in [Appendix A.1](#).

The TOE also includes the TOE documentation providing information for installing, configuring, and maintaining the evaluated configuration.

- Apple macOS 13 Ventura Common Criteria Configuration Guide, Version 1.1 [CCGUIDE] [📄](#)

## 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:

- Start-up and shut-down of the 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)

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 corecrypto v13.0 cryptographic libraries for performing user space, kernel space, and SEP cryptographic operations. In addition, it uses a software noise source for entropy generation. The TOE implements TLS 1.2 for secure communications with remote servers.

The Bluetooth hardware implements the bulk AES-CCM-128 cryptographic functionality used when connecting to Bluetooth devices.

### **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 an asymmetric key stored in OE-protected storage

The TOE will lock out user accounts after a defined number of unsuccessful authentication attempts has been met.

The TOE supports Bluetooth Secure Simple Pairing (SSP). It requires user authorization and mutual authentication during pairing. It also discards pairing attempts and session initialization from Bluetooth devices to which an active session preexists. The TOE requires explicit user authorization when pairing with an untrusted device.

### **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 will have limited privileges.

### **1.5.3.6 Protection of the TSF**

The TOE implements the following protection of TSF data functions:

- Access controls
- 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**

The TOE displays an advisory warning message regarding unauthorized use of the OS prior to establishment of a user session.

### **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 for both BR/EDR and LE and terminates the connection if the connected device stops encrypting.

## **1.5.4 TOE operational environment**

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

**Table 1: TOE operational environment**

<b>Component</b>	<b>Description</b>
Hardware platform	See Appendix A.1
Apple Update Server	Server that allows the TOE to download updates

## 2 CC Conformance Claim

This Security Target is CC Part 2 extended and CC Part 3 extended.

This Security Target claims conformance to the following Protection Profiles and PP packages:

- [\[CFG\\_GPOS-BT\\_V1.0\]](#): PP-Configuration for General Purpose Operating Systems and Bluetooth. Version 1.0 as of 2021-04-15; exact conformance.
- [\[OSPPv4.2.1\]](#): Protection Profile for General Purpose Operating Systems. Version 4.2.1 as of 2019-04-22; exact conformance.
- [\[BT\]](#): PP-Module for Bluetooth. Version 1.0 as of 2021-04-15; exact conformance.

Common Criteria [CC] version 3.1 revision 5 is the basis for this conformance claim.

### 2.1 Protection Profile Tailoring and Additions

#### 2.1.1 Protection Profile for General Purpose Operating Systems ([OSPPv4.2.1])

This document claims conformance to the following OSPP Use Case:

- [Use Case 1] End User Devices: The TOE provides a platform for end user device such as desktops, laptops, convertibles, and tablets.

Table 2 contains the NIAP Technical Decisions (TDs) for this protection profile at the time of the evaluation and a statement of applicability to the evaluation.

**Table 2: NIAP TDs for OSPP**

NIAP TD	TD description	Applicable?	Non-applicability rationale
<a href="#">TD0715</a>	Updates to FIA_X509_EXT.1 for Exception Processing and Test Conditions	Yes	
<a href="#">TD0680</a>	OS 4.2.1 Conformance Claims section updated to allow for MOD_WLAN_CLI_v1.0	No	This evaluation does not include MOD_WLAN_CLI_V1.0.
<a href="#">TD0649</a>	Conformance claims for OS PP v4.2.1	Yes	
<a href="#">TD0630</a>	FCS_COP.1 requirements for Secure Shell	No	This evaluation does not include Secure Shell.
<a href="#">TD0600</a>	Conformance claim sections updated to allow for MOD_VPNC_V2.3	No	This evaluation does not include MOD_VPNC_V2.3.
<a href="#">TD0578</a>	SHA-1 is no longer mandatory	Yes	
<a href="#">TD0501</a>	Cryptographic selections and updates for OS PP	Yes	
<a href="#">TD0493</a>	X.509v3 certificates when using digital signatures for Boot Integrity	Yes	
<a href="#">TD0463</a>	Clarification for FPT_TUD_EXT	Yes	
<a href="#">TD0441</a>	Updated TLS Ciphersuites for OS PP	Yes	
<a href="#">TD0386</a>	Platform-Provided Verification of Update	Yes	
<a href="#">TD0365</a>	FCS_CKM_EXT.4 selections	Yes	

## 2.1.2 PP-Module for Bluetooth ([BT])

This document claims conformance to the following Bluetooth Use Case:

- [Use Case 1] General-Purpose Operating System: The bluetooth functionality provided by the TOE is part of the general-purpose operating system itself. No standalone third-party applications are necessary to be installed.

Table 3 contains the NIAP Technical Decisions (TDs) for this PP-Module at the time of the evaluation and a statement of applicability to the evaluation.

**Table 3: NIAP TDs for BT**

NIAP TD	TD description	Applicable?	Non-applicability rationale
<a href="#">TD0707</a>	Formatting corrections for MOD_BT_V1.0	Yes	
<a href="#">TD0685</a>	BT missing multiple SFR-to-Obj mappings	Yes	
<a href="#">TD0671</a>	Bluetooth PP-Module updated to allow for new PP and PP-Module Versions	Yes	
<a href="#">TD0650</a>	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 MOD_VPNC_V2.4.
<a href="#">TD0645</a>	Bluetooth audit details	Yes	
<a href="#">TD0640</a>	Handling BT devices that do not support encryption	Yes	
<a href="#">TD0600</a>	Conformance claim sections updated to allow for MOD_VPNC_V2.3	No	This evaluation does not include MOD_VPNC_V2.3.

## 3 Security Problem Definition

### 3.1 Threat Environment

#### 3.1.1 Threats countered by the TOE

##### T.NETWORK\_ATTACK

**PP Origin:** OSPP, BT

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

##### T.NETWORK\_EAVESDROP

**PP Origin:** OSPP, BT

An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between applications and services that are running on or part of the OS.

##### T.LOCAL\_ATTACK

**PP Origin:** OSPP

An attacker may compromise applications running on the OS. The compromised application may provide maliciously formatted input to the OS through a variety of channels including unprivileged system calls and messaging via the file system.

##### T.LIMITED\_PHYSICAL\_ACCESS

**PP Origin:** OSPP

An attacker may attempt to access data on the OS while having a limited amount of time with the physical device.

### 3.2 Assumptions

#### A.PLATFORM

**PP Origin:** OSPP

The OS relies upon a trustworthy computing platform for its execution. This underlying platform is out of scope of this PP.

#### A.PROPER\_USER

**PP Origin:** OSPP

The user of the OS is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy. At the same time, malicious software could act as the user, so requirements which confine malicious subjects are still in scope.

#### A.PROPER\_ADMIN

**PP Origin:** OSPP

The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

## 4 Security Objectives

### 4.1 Objectives for the TOE

#### O.ACCOUNTABILITY

**PP Origin:** OSPP

Conformant OSEs ensure that information exists that allows administrators to discover unintentional issues with the configuration and operation of the operating system and discover its cause. Gathering event information and immediately transmitting it to another system can also enable incident response in the event of system compromise.

#### O.INTEGRITY

**PP Origin:** OSPP

Conformant OSEs ensure the integrity of their update packages. OSEs are seldom if ever shipped without errors, and the ability to deploy patches and updates with integrity is critical to enterprise network security. Conformant OSEs provide execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems.

#### O.MANAGEMENT

**PP Origin:** OSPP

To facilitate management by users and the enterprise, conformant OSEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration and application execution control.

#### O.PROTECTED\_STORAGE

**PP Origin:** OSPP

To address the issue of loss of confidentiality of credentials in the event of loss of physical control of the storage medium, conformant OSEs provide data-at-rest protection for credentials. Conformant OSEs also provide access controls which allow users to keep their files private from other users of the same system.

#### O.PROTECTED\_COMMS

**PP Origin:** OSPP, BT

To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant OSEs provide mechanisms to create trusted channels for CSP and sensitive data. Both CSP and sensitive data should not be exposed outside of the platform.

### 4.2 Objectives for the Operational Environment

#### OE.PLATFORM

**PP Origin:** OSPP

The OS relies on being installed on trusted hardware.

#### OE.PROPER\_USER

**PP Origin:** OSPP

The user of the OS is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy. Standard user accounts are provisioned in accordance with the least privilege model. Users requiring higher levels of access should have a separate account dedicated for that use.

**OE.PROPER\_ADMIN****PP Origin:** OSPP

The administrator of the OS is not careless, willfully negligent or hostile, and administers the OS within compliance of the applied enterprise security policy.

**4.3 Security Objectives Rationale****4.3.1 Coverage**

The following table provides a mapping of TOE objectives to threats and policies, showing that each objective counters or enforces at least one threat or policy, respectively.

**Table 4: Mapping of security objectives to threats and policies**

Objective	Threats / OSPs
O.ACCOUNTABILITY	T.NETWORK_ATTACK T.LOCAL_ATTACK
O.INTEGRITY	T.NETWORK_ATTACK T.LOCAL_ATTACK
O.MANAGEMENT	T.NETWORK_ATTACK T.NETWORK_EAVESDROP
O.PROTECTED_STORAGE	T.LIMITED_PHYSICAL_ACCESS
O.PROTECTED_COMMS	T.NETWORK_ATTACK T.NETWORK_EAVESDROP

The following table provides a mapping of the objectives for the Operational Environment to assumptions, threats and policies, showing that each objective holds, counters or enforces at least one assumption, threat or policy, respectively.

**Table 5: Mapping of security objectives for the Operational Environment to assumptions, threats and policies**

Objective	Assumptions / Threats / OSPs
OE.PLATFORM	A.PLATFORM
OE.PROPER_USER	A.PROPER_USER
OE.PROPER_ADMIN	A.PROPER_ADMIN

**4.3.2 Sufficiency**

The following rationale provides justification that the security objectives are suitable to counter each individual threat and that each security objective tracing back to a threat, when achieved, actually contributes to the removal, diminishing or mitigation of that threat.

**Table 6: Sufficiency of objectives countering threats**

Threat	Rationale for security objectives
T.NETWORK_ATTACK	<p>[OSPP] The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides for integrity of transmitted data.</p> <p>[OSPP] The threat T.NETWORK_ATTACK is countered by O.INTEGRITY as this provides for integrity of software that is installed onto the system from the network.</p>



Threat	Rationale for security objectives
	<p>[OSPP] The threat T.NETWORK_ATTACK is countered by O.MANAGEMENT as this provides for the ability to configure the OS to defend against network attack.</p> <p>[OSPP] The threat T.NETWORK_ATTACK is countered by O.ACCOUNTABILITY as this provides a mechanism for the OS to report behavior that may indicate a network attack has occurred.</p> <p>[BT] The threat T.NETWORK_ATTACK is countered by O.PROTECTED_COMMS as this provides the capability to communicate using Bluetooth as a means to maintain the confidentiality of data that are transmitted outside of the TOE.</p>
T.NETWORK_EAVESDROP	<p>[OSPP] The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides for confidentiality of transmitted data.</p> <p>[OSPP] The threat T.NETWORK_EAVESDROP is countered by O.MANAGEMENT as this provides for the ability to configure the OS to protect the confidentiality of its transmitted data.</p> <p>[BT] The threat T.NETWORK_EAVESDROP is countered by O.PROTECTED_COMMS as this provides the capability to communicate using Bluetooth as a means to maintain the confidentiality of data that are transmitted outside of the TOE.</p>
T.LOCAL_ATTACK	<p>[OSPP] The objective O.INTEGRITY protects against the use of mechanisms that weaken the TOE with regard to attack by other software on the platform.</p> <p>[OSPP] The objective O.ACCOUNTABILITY protects against local attacks by providing a mechanism to report behavior that may indicate a local attack is occurring or has occurred.</p>
T.LIMITED_PHYSICAL_ACCESS	<p>[OSPP] The objective O.PROTECTED_STORAGE protects against unauthorized attempts to access physical storage used by the TOE.</p>

The following rationale provides justification that the security objectives for the environment are suitable to cover each individual assumption, that each security objective for the environment that traces back to an assumption about the environment of use of the TOE, when achieved, actually contributes to the environment achieving consistency with the assumption, and that if all security objectives for the environment that trace back to an assumption are achieved, the intended usage is supported.

**Table 7: Sufficiency of objectives holding assumptions**

Assumption	Rationale for security objectives
A.PLATFORM	[OSPP] The operational environment objective OE.PLATFORM is realized through A.PLATFORM.
A.PROPER_USER	[OSPP] The operational environment objective OE.PROPER_USER is realized through A.PROPER_USER.
A.PROPER_ADMIN	[OSPP] The operational environment objective OE.PROPER_ADMIN is realized through A.PROPER_ADMIN.

## 5 Extended Components Definition

The extended components definitions are defined in the documents specified in [Section 2 "CC Conformance Claim"](#).

## 6 Security Requirements

### 6.1 TOE Security Functional Requirements

The table below summarizes the SFRs for the TOE and the operations performed on the components according to CC part 1. Operations in the SFRs use the following convention:

- Iterations (Iter.) are identified by appending a suffix to the original SFR.
- Refinements (Ref.) added to the text are shown in *italic text*, deletions are shown as ~~text~~.
- Assignments (Ass.) are shown in **bold text**.
- Selections (Sel.) are shown in **bold text**.

Table 8: SFRs for the TOE

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FAU - Security audit	FAU_GEN.1 Audit Data Generation (Refined)		OSPPv4.2.1	Yes	No	Yes	Yes
	FAU_GEN.1/BT Audit Data Generation (Bluetooth)	FAU_GEN.1	BT	Yes	Yes	No	Yes
FCS - Cryptographic support	FCS_CKM.1 Cryptographic Key Generation (Refined)		OSPPv4.2.1	No	No	No	Yes
	FCS_CKM.2 Cryptographic Key Establishment (Refined)		OSPPv4.2.1	No	No	No	Yes
	FCS_CKM_EXT.4 Cryptographic Key Destruction		OSPPv4.2.1	No	No	No	Yes
	FCS_CKM_EXT.8 Bluetooth Key Generation		BT	No	No	Yes	No
	FCS_COP.1(1) Cryptographic Operation - Encryption/Decryption (Refined)	FCS_COP.1	OSPPv4.2.1	Yes	No	No	Yes
	FCS_COP.1(2) Cryptographic Operation - Hashing (Refined)	FCS_COP.1	OSPPv4.2.1	Yes	No	No	Yes
	FCS_COP.1(3) Cryptographic Operation - Signing (Refined)	FCS_COP.1	OSPPv4.2.1	Yes	No	No	Yes
	FCS_COP.1(4) Cryptographic Operation - Keyed-Hash Message Authentication (Refined)	FCS_COP.1	OSPPv4.2.1	Yes	No	Yes	Yes
	FCS_RBG_EXT.1 Random Bit Generation		OSPPv4.2.1	No	No	No	Yes
	FCS_STO_EXT.1 Storage of Sensitive Data		OSPPv4.2.1	No	No	No	No
	FCS_TLSC_EXT.1 TLS Client Protocol		OSPPv4.2.1	No	No	No	Yes
	FCS_TLSC_EXT.2 TLS Client Protocol		OSPPv4.2.1	No	No	No	Yes
	FCS_TLSC_EXT.4 TLS Client Protocol		OSPPv4.2.1	No	No	No	No

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
FDP - User data protection	FDP_ACF_EXT.1 Access Controls for Protecting User Data		OSPPv4.2.1	No	No	No	No
FIA - Identification and authentication	FIA_AFL.1 Authentication failure handling (Refined)		OSPPv4.2.1	No	No	Yes	Yes
	FIA_BLT_EXT.1 Bluetooth User Authorization		BT	No	No	No	No
	FIA_BLT_EXT.2 Bluetooth Mutual Authentication		BT	No	No	No	No
	FIA_BLT_EXT.3 Rejection of Duplicate Bluetooth Connections		BT	No	No	No	No
	FIA_BLT_EXT.4 Secure Simple Pairing		BT	No	No	No	No
	FIA_BLT_EXT.6 Trusted Bluetooth Device User Authorization		BT	No	No	Yes	No
	FIA_BLT_EXT.7 Untrusted Bluetooth Device User Authorization		BT	No	No	Yes	No
	FIA_UAU.5 Multiple Authentication Mechanisms (Refined)		OSPPv4.2.1	No	No	Yes	Yes
	FIA_X509_EXT.1 X.509 Certificate Validation		OSPPv4.2.1	No	No	No	Yes
	FIA_X509_EXT.2 X.509 Certificate Authentication		OSPPv4.2.1	No	No	No	Yes
FMT - Security management	FMT_MOF_EXT.1 Management of security functions behavior		OSPPv4.2.1	Yes	No	No	No
	FMT_MOF_EXT.1/BT Management of Security Functions Behavior	FMT_MOF_EXT.1	BT	Yes	No	No	No
	FMT_SMF_EXT.1 Specification of Management Functions		OSPPv4.2.1	Yes	No	Yes	Yes
	FMT_SMF_EXT.1/BT Specification of Management Functions	FMT_SMF_EXT.1	BT	Yes	No	No	No
FPT - Protection of the TSF	FPT_ACF_EXT.1 Access controls		OSPPv4.2.1	No	No	Yes	No
	FPT_ASLR_EXT.1 Address Space Layout Randomization		OSPPv4.2.1	No	No	Yes	Yes
	FPT_SBOP_EXT.1 Stack Buffer Overflow Protection		OSPPv4.2.1	No	No	No	Yes
	FPT_TST_EXT.1 Boot Integrity		OSPPv4.2.1	No	No	No	Yes
	FPT_TUD_EXT.1 Trusted Update		OSPPv4.2.1	No	No	No	Yes
	FPT_TUD_EXT.2 Trusted Update for Application Software		OSPPv4.2.1	No	No	No	No

Security functional class	Security functional requirement	Base security functional component	Source	Operations			
				Iter.	Ref.	Ass.	Sel.
	FPT_W^X_EXT.1 Write XOR Execute Memory Pages		OSPPv4.2.1	No	No	Yes	No
FTA - TOE access	FTA_TAB.1 Default TOE access banners		OSPPv4.2.1	No	No	No	No
FTP - Trusted path/channels	FTP_BLT_EXT.1 Bluetooth Encryption		BT	No	No	No	Yes
	FTP_BLT_EXT.2 Persistence of Bluetooth Encryption		BT	No	No	No	Yes
	FTP_BLT_EXT.3/BR Bluetooth Encryption Parameters (BR/EDR)	FTP_BLT_EXT.3	BT	Yes	No	Yes	No
	FTP_BLT_EXT.3/LE Bluetooth Encryption Parameters (LE)	FTP_BLT_EXT.3	BT	Yes	No	Yes	No
	FTP_ITC_EXT.1 Trusted channel communication		OSPPv4.2.1	No	No	Yes	Yes
	FTP_TRP.1 Trusted Path		OSPPv4.2.1	No	No	No	Yes

## 6.1.1 Security audit (FAU)

### 6.1.1.1 FAU\_GEN.1 Audit Data Generation (Refined)

PP Origin: OSPP

#### 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: [TSS for FAU\\_GEN.1](#)

### 6.1.1.2 FAU\_GEN.1/BT Audit Data Generation (Bluetooth)

PP Origin: BT

Applied TDs: [TD0645](#) [TD0707](#)

### FAU\_GEN.1.1/BT

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

- a. Start-up and shutdown of the audit functions
- b. All auditable events for the not specified level of audit
- c. Specifically defined auditable events in the Auditable Events table ( [Table 9](#) ).

**Table 9: 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.	<b>Complete</b> BD_ADDR and <b>no other information</b> . Bluetooth profile. Identity of local service with <b>service ID</b> .
FIA_BLT_EXT.2	Initiation of Bluetooth connection.	<b>Complete</b> BD_ADDR and <b>no other information</b> .
	Failure of Bluetooth connection.	Reason for failure.
FIA_BLT_EXT.3 (optional)	Duplicate connection attempt.	BD_ADDR of connection attempt.
FIA_BLT_EXT.4	None.	
FIA_BLT_EXT.5 (if claimed)	None.	
FIA_BLT_EXT.6	None.	
FIA_BLT_EXT.7	None.	
FTP_BLT_EXT.1	None.	
FTP_BLT_EXT.2	None.	
FTP_BLT_EXT.3/BR	None.	
FTP_BLT_EXT.3/LE (if claimed)	None.	

**Application Note:** *FIA\_BLT\_EXT.3 is crossed out in [Table 9](#) because the rejection is performed at the HCI layer. FIA\_BLT\_EXT.5 is crossed out in [Table 9](#) because it is not claimed by this ST.*

### 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 (*Table 9*).

**TSS Link:** [TSS for FAU\\_GEN.1/BT](#)

## 6.1.2 Cryptographic support (FCS)

### 6.1.2.1 FCS\_CKM.1 Cryptographic Key Generation (Refined)

**PP Origin:** OSPP

**Applied TDs:** [TD0501](#)

#### 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 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3**
- **ECC schemes using "NIST curves" P-256, P-384 and P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4**

**TSS Link:** [TSS for FCS\\_CKM.1](#)

### 6.1.2.2 FCS\_CKM.2 Cryptographic Key Establishment (Refined)

**PP Origin:** OSPP

**Applied TDs:** [TD0501](#)

#### 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:** [TSS for FCS\\_CKM.2](#)

### 6.1.2.3 FCS\_CKM\_EXT.4 Cryptographic Key Destruction

**PP Origin:** OSPP

**Applied TDs:** [TD0365](#)

#### FCS\_CKM\_EXT.4.1

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

- **For volatile memory, the destruction shall be executed by a**
  - **single overwrite consisting of zeroes**
- **For non-volatile memory that consists of**
  - **destruction of all key encrypting keys protecting the target key according to FCS\_CKM\_EXT.4.1, where none of the KEKs protecting the target key are derived**

#### FCS\_CKM\_EXT.4.2

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

TSS Link: [TSS for FCS\\_CKM\\_EXT.4](#)

### 6.1.2.4 FCS\_CKM\_EXT.8 Bluetooth Key Generation

PP Origin: *BT*

#### FCS\_CKM\_EXT.8.1

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

TSS Link: [TSS for FCS\\_CKM\\_EXT.8](#)

### 6.1.2.5 FCS\_COP.1(1) Cryptographic Operation - Encryption/Decryption (Refined)

PP Origin: *OSPP*

#### FCS\_COP.1.1(1)

The OS shall perform encryption/decryption services for data in accordance with a specified cryptographic algorithm

- **AES-CBC (as defined in NIST SP 800-38A)**
- and
- **AES-GCM (as defined in NIST SP 800-38D)**
  - **AES-CCM (as defined in NIST SP 800-38C)**
- and cryptographic key sizes **128-bit, 256-bit**.

TSS Link: [TSS for FCS\\_COP.1\(1\)](#)

### 6.1.2.6 FCS\_COP.1(2) Cryptographic Operation - Hashing (Refined)

PP Origin: *OSPP*

Applied TDs: [TD0578](#)

#### FCS\_COP.1.1(2)

The OS shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm

- **SHA-1**
- **SHA-256**
- **SHA-384**
- **SHA-512**



and message digest sizes

- **160 bits**
- **256 bits**
- **384 bits**
- **512 bits**

that meet the following: FIPS Pub 180-4.

**TSS Link:** [TSS for FCS\\_COP.1\(2\)](#)

### 6.1.2.7 FCS\_COP.1(3) Cryptographic Operation - Signing (Refined)

**PP Origin:** OSPP

#### FCS\_COP.1.1(3)

The OS shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm

- **RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 4**
- **ECDSA schemes using "NIST curves" P-256, P-384 and P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5**

**TSS Link:** [TSS for FCS\\_COP.1\(3\)](#)

### 6.1.2.8 FCS\_COP.1(4) Cryptographic Operation - Keyed-Hash Message Authentication (Refined)

**PP Origin:** OSPP

#### FCS\_COP.1.1(4)

The OS shall perform keyed-hash message authentication services in accordance with a specified cryptographic algorithm **SHA-1, SHA-256, SHA-384, SHA-512** with key sizes **160 bits, 256 bits, 384 bits, 512 bits** and message digest sizes **160 bits, 256 bits, 384 bits, 512 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:** [TSS for FCS\\_COP.1\(4\)](#)

### 6.1.2.9 FCS\_RBG\_EXT.1 Random Bit Generation

**PP Origin:** OSPP

#### 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** 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:** [TSS for FCS\\_RBG\\_EXT.1](#)

### 6.1.2.10 FCS\_STO\_EXT.1 Storage of Sensitive Data

**PP Origin:** OSPP

#### 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:** [TSS for FCS\\_STO\\_EXT.1](#)

### 6.1.2.11 FCS\_TLSC\_EXT.1 TLS Client Protocol

**PP Origin:** OSPP

**Applied TDs:** [TD0441](#)

#### FCS\_TLSC\_EXT.1.1

The OS shall implement TLS 1.2 (RFC 5246) supporting the following cipher suites:

- **TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 5246**
- **TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 5246**
- **TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5288**
- **TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5288**
- **TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289**
- **TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289**
- **TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289**
- **TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289**

#### FCS\_TLSC\_EXT.1.2

The OS shall verify that the presented identifier matches the reference identifier according to RFC 6125.

#### FCS\_TLSC\_EXT.1.3

The OS shall only establish a trusted channel if the peer certificate is valid.

**TSS Link:** [TSS for FCS\\_TLSC\\_EXT.1](#)

### 6.1.2.12 FCS\_TLSC\_EXT.2 TLS Client Protocol

**PP Origin:** OSPP

#### FCS\_TLSC\_EXT.2.1

The OS shall present the Supported Groups Extension in the Client Hello with the following supported groups: **secp256r1, secp384r1, secp521r1**.

**TSS Link:** [TSS for FCS\\_TLSC\\_EXT.2](#)

### 6.1.2.13 FCS\_TLSC\_EXT.4 TLS Client Protocol

**PP Origin:** OSPP

#### FCS\_TLSC\_EXT.4.1

The OS shall support mutual authentication using X.509v3 certificates.

**TSS Link:** [TSS for FCS\\_TLSC\\_EXT.4](#)

### 6.1.3 User data protection (FDP)

#### 6.1.3.1 FDP\_ACF\_EXT.1 Access Controls for Protecting User Data

**PP Origin:** OSPP

##### 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:** [TSS for FDP\\_ACF\\_EXT.1](#)

### 6.1.4 Identification and authentication (FIA)

#### 6.1.4.1 FIA\_AFL.1 Authentication failure handling (Refined)

**PP Origin:** OSPP

##### 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**.

##### FIA\_AFL.1.2

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

**TSS Link:** [TSS for FIA\\_AFL.1](#)

#### 6.1.4.2 FIA\_BLT\_EXT.1 Bluetooth User Authorization

**PP Origin:** BT

##### FIA\_BLT\_EXT.1.1

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

**TSS Link:** [TSS for FIA\\_BLT\\_EXT.1](#)

#### 6.1.4.3 FIA\_BLT\_EXT.2 Bluetooth Mutual Authentication

**PP Origin:** BT

##### 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:** [TSS for FIA\\_BLT\\_EXT.2](#)

#### **6.1.4.4 FIA\_BLT\_EXT.3 Rejection of Duplicate Bluetooth Connections**

**PP Origin:** *BT*

##### **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:** [TSS for FIA\\_BLT\\_EXT.3](#)

#### **6.1.4.5 FIA\_BLT\_EXT.4 Secure Simple Pairing**

**PP Origin:** *BT*

##### **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:** [TSS for FIA\\_BLT\\_EXT.4](#)

#### **6.1.4.6 FIA\_BLT\_EXT.6 Trusted Bluetooth Device User Authorization**

**PP Origin:** *BT*

##### **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:** [TSS for FIA\\_BLT\\_EXT.6](#)

#### **6.1.4.7 FIA\_BLT\_EXT.7 Untrusted Bluetooth Device User Authorization**

**PP Origin:** *BT*

##### **FIA\_BLT\_EXT.7.1**

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

**TSS Link:** [TSS for FIA\\_BLT\\_EXT.7](#)

#### **6.1.4.8 FIA\_UAU.5 Multiple Authentication Mechanisms (Refined)**

**PP Origin:** *OSPP*

**Applied TDs:** [TD0649](#)

##### **FIA\_UAU.5.1**

The OS shall provide the following authentication mechanisms:

- **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**

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:** [TSS for FIA\\_UAU.5](#)

### 6.1.4.9 FIA\_X509\_EXT.1 X.509 Certificate Validation

**PP Origin:** OSPP

**Applied TDs:** [TD0715](#)

#### 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 caSigning purpose in 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 ECU 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 ECU 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 ECU 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 ECU field.**

#### 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:** [TSS for FIA\\_X509\\_EXT.1](#)

## 6.1.4.10 FIA\_X509\_EXT.2 X.509 Certificate Authentication

PP Origin: OSPP

### FIA\_X509\_EXT.2.1

The OS shall use X.509v3 certificates as defined by RFC 5280 to support authentication for TLS and HTTPS connections.

TSS Link: [TSS for FIA\\_X509\\_EXT.2](#)

## 6.1.5 Security management (FMT)

### 6.1.5.1 FMT\_MOF\_EXT.1 Management of security functions behavior

PP Origin: OSPP

#### 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: [TSS for FMT\\_MOF\\_EXT.1](#)

### 6.1.5.2 FMT\_MOF\_EXT.1/BT Management of Security Functions Behavior

PP Origin: BT

#### FMT\_MOF\_EXT.1.1/BT

The OS shall restrict the ability to perform the function indicated in the "Administrator" column in FMT\_SMF\_EXT.1.1/BT to the administrator.

TSS Link: [TSS for FMT\\_MOF\\_EXT.1/BT](#)

### 6.1.5.3 FMT\_SMF\_EXT.1 Specification of Management Functions

PP Origin: OSPP, BT

#### FMT\_SMF\_EXT.1.1

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

Table 10: Management functions (OSPP)

#	Management Function	Administrator	User
1	Enable/disable <b>screen lock</b>	M	X
2	Configure <b>screen lock</b> inactivity timeout	M	X
3	Configure local audit storage capacity	X	-
4	Configure minimum password length	X	-
5	Configure minimum number of special characters in password	X	-
6	Configure minimum number of numeric characters in password	-	-
7	Configure minimum number of uppercase characters in password	-	-

#	Management Function	Administrator	User
8	Configure minimum number of lowercase characters in password	-	-
9	Configure lockout policy for unsuccessful authentication attempts through <b>timeouts between attempts, limiting number of attempts during a time period</b>	-	-
10	Configure host-based firewall	X	-
11	Configure name/address of directory server with which to bind	-	-
12	Configure name/address of remote management server from which to receive management settings	X	-
13	Configure name/address of audit/logging server to which to send audit/logging records	X	-
14	Configure audit rules	X	-
15	Configure name/address of network time server	X	-
16	Enable/disable automatic software update	X	-
17	Configure Wi-Fi interface	X	X
18	Enable/disable Bluetooth interface	-	X
19	Enable/disable <b>no other external interfaces</b>	-	-
20	<b>No other management functions to be provided by the TSF</b>	-	-

**Application Note:** *M*—Mandatory support by the specified role. *X*—Supported by the specified role. Grey/Hyphen—Not supported by the specified role.

**TSS Link:** [TSS for FMT\\_SMF\\_EXT.1](#)

## 6.1.5.4 FMT\_SMF\_EXT.1/BT Specification of Management Functions

**PP Origin:** *BT*

### FMT\_SMF\_EXT.1.1/BT

The OS shall be capable of performing the following Bluetooth management functions:

**Table 11: Management functions (Bluetooth)**

#	Management Function	Administrator	User
BT-1	Configure the Bluetooth trusted channel. <ul style="list-style-type: none"> <li>Disable/enable the Discoverable (for BR/EDR) and Advertising (for LE) modes;</li> </ul>	M	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;	-	-

#	Management Function	Administrator	User
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:** *M*—Mandatory support by the specified role. *X*—Supported by the specified role. Grey/Hyphen—Not supported by the specified role.

**TSS Link:** [TSS for FMT\\_SMF\\_EXT.1/BT](#)

## 6.1.6 Protection of the TSF (FPT)

### 6.1.6.1 FPT\_ACF\_EXT.1 Access controls

**PP Origin:** OSPP

#### FPT\_ACF\_EXT.1.1

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

- a) Kernel and its drivers/modules
- b) Security audit logs
- c) Shared libraries
- d) System executables
- e) System configuration files
- f) **TSF-data**
- g) **Applications**

#### FPT\_ACF\_EXT.1.2

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

- a) Security audit logs
- b) System-wide credential repositories
- c) **no other objects**

**TSS Link:** [TSS for FPT\\_ACF\\_EXT.1](#)

### 6.1.6.2 FPT\_AS LR\_EXT.1 Address Space Layout Randomization

**PP Origin:** OSPP

#### FPT\_AS LR\_EXT.1.1

The OS shall always randomize process address space memory locations with **16** bits of entropy except for **no exceptions**.

**TSS Link:** [TSS for FPT\\_AS LR\\_EXT.1](#)

### 6.1.6.3 FPT\_SBOP\_EXT.1 Stack Buffer Overflow Protection

**PP Origin:** OSPP



#### FPT\_SBOP\_EXT.1.1

The OS shall **employ stack-based buffer overflow protections**.

**TSS Link:** [TSS for FPT\\_SBOP\\_EXT.1](#)

### 6.1.6.4 FPT\_TST\_EXT.1 Boot Integrity

**PP Origin:** OSPP

**Applied TDs:** [TD0493](#)

#### 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:** [TSS for FPT\\_TST\\_EXT.1](#)

### 6.1.6.5 FPT\_TUD\_EXT.1 Trusted Update

**PP Origin:** OSPP

**Applied TDs:** [TD0386](#), [TD0463](#)

#### 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\(3\)](#) 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\(3\)](#).

**TSS Link:** [TSS for FPT\\_TUD\\_EXT.1](#)

### 6.1.6.6 FPT\_TUD\_EXT.2 Trusted Update for Application Software

**PP Origin:** OSPP

**Applied TDs:** [TD0463](#)

#### 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\(3\)](#) 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\(3\)](#) prior to installation.

**TSS Link:** [TSS for FPT\\_TUD\\_EXT.2](#)

## 6.1.6.7 FPT\_W^X\_EXT.1 Write XOR Execute Memory Pages

PP Origin: OSPP

### FPT\_W^X\_EXT.1.1

The OS shall prevent allocation of any memory region with both write and execute permissions except for

- **On Apple silicon Macs: Safari, Perl, Python, JavaScript**
- **On "Intel with T2" Macs: Applications not marked with hardened execution in their binary image**

TSS Link: *TSS for FPT\_W^X\_EXT.1*

## 6.1.7 TOE access (FTA)

### 6.1.7.1 FTA\_TAB.1 Default TOE access banners

PP Origin: OSPP

#### 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: *TSS for FTA\_TAB.1*

## 6.1.8 Trusted path/channels (FTP)

### 6.1.8.1 FTP\_BLT\_EXT.1 Bluetooth Encryption

PP Origin: BT

#### 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: *TSS for FTP\_BLT\_EXT.1*

### 6.1.8.2 FTP\_BLT\_EXT.2 Persistence of Bluetooth Encryption

PP Origin: BT

#### FTP\_BLT\_EXT.2.1

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

TSS Link: *TSS for FTP\_BLT\_EXT.2*

### 6.1.8.3 FTP\_BLT\_EXT.3/BR Bluetooth Encryption Parameters (BR/EDR)

PP Origin: BT

Applied TDs: [TD0640](#)

#### 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: [TSS for FTP\\_BLT\\_EXT.3/BR](#)

### 6.1.8.4 FTP\_BLT\_EXT.3/LE Bluetooth Encryption Parameters (LE)

PP Origin: *BT*

#### 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: [TSS for FTP\\_BLT\\_EXT.3/LE](#)

### 6.1.8.5 FTP\_ITC\_EXT.1 Trusted channel communication

PP Origin: *OSPP*

Applied TDs: [TD0649](#)

#### FTP\_ITC\_EXT.1.1

The OS shall use

- **TLS as conforming to** [FCS\\_TLSC\\_EXT.1](#)

to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: **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: [TSS for FTP\\_ITC\\_EXT.1](#)

### 6.1.8.6 FTP\_TRP.1 Trusted Path

PP Origin: *OSPP*

#### 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 all remote administrative actions.

TSS Link: [TSS for FTP\\_TRP.1](#)

## 6.2 Security Functional Requirements Rationale

### 6.2.1 Coverage

The following table provides a mapping of SFR to the security objectives, showing that each security functional requirement addresses at least one security objective.

Table 12: Mapping of security functional requirements to security objectives

Security functional requirements	Objectives
FAU_GEN.1	O.ACCOUNTABILITY
FAU_GEN.1/BT	O.ACCOUNTABILITY, O.INTEGRITY
FCS_CKM.1	O.PROTECTED_COMMS
FCS_CKM.2	O.PROTECTED_COMMS
FCS_CKM_EXT.4	O.PROTECTED_COMMS
FCS_CKM_EXT.8	O.PROTECTED_COMMS
FCS_COP.1(1)	O.PROTECTED_COMMS, O.PROTECTED_STORAGE
FCS_COP.1(2)	O.INTEGRITY, O.PROTECTED_COMMS
FCS_COP.1(3)	O.INTEGRITY, O.PROTECTED_COMMS
FCS_COP.1(4)	O.INTEGRITY, O.PROTECTED_COMMS
FCS_RBG_EXT.1	O.PROTECTED_COMMS, O.PROTECTED_STORAGE
FCS_STO_EXT.1	O.PROTECTED_STORAGE
FCS_TLSC_EXT.1	O.PROTECTED_COMMS
FCS_TLSC_EXT.2	O.PROTECTED_COMMS
FCS_TLSC_EXT.4	O.PROTECTED_COMMS
FDP_ACF_EXT.1	O.PROTECTED_STORAGE
FIA_AFL.1	O.INTEGRITY
FIA_BLT_EXT.1	O.PROTECTED_COMMS
FIA_BLT_EXT.2	O.PROTECTED_COMMS
FIA_BLT_EXT.3	O.PROTECTED_COMMS
FIA_BLT_EXT.4	O.PROTECTED_COMMS
FIA_BLT_EXT.6	O.PROTECTED_COMMS
FIA_BLT_EXT.7	O.PROTECTED_COMMS
FIA_UAU.5	O.INTEGRITY
FIA_X509_EXT.1	O.INTEGRITY,

Security functional requirements	Objectives
	O.PROTECTED_COMMS
FIA_X509_EXT.2	O.PROTECTED_COMMS
FMT_MOF_EXT.1	O.MANAGEMENT
FMT_MOF_EXT.1/BT	O.MANAGEMENT
FMT_SMF_EXT.1	O.MANAGEMENT
FMT_SMF_EXT.1/BT	O.MANAGEMENT
FPT_ACF_EXT.1	O.INTEGRITY
FPT_ASLR_EXT.1	O.INTEGRITY
FPT_SBOP_EXT.1	O.INTEGRITY
FPT_TST_EXT.1	O.INTEGRITY
FPT_TUD_EXT.1	O.INTEGRITY
FPT_TUD_EXT.2	O.INTEGRITY
FPT_W^X_EXT.1	O.INTEGRITY
FTA_TAB.1	O.MANAGEMENT
FTP_BLT_EXT.1	O.PROTECTED_COMMS
FTP_BLT_EXT.2	O.PROTECTED_COMMS
FTP_BLT_EXT.3/BR	O.PROTECTED_COMMS
FTP_BLT_EXT.3/LE	O.PROTECTED_COMMS
FTP_ITC_EXT.1	O.ACCOUNTABILITY, O.INTEGRITY, O.PROTECTED_COMMS
FTP_TRP.1	O.MANAGEMENT

## 6.2.2 Sufficiency

The following rationale provides justification for each security objective for the TOE, showing that the security functional requirements are suitable to meet and achieve the security objectives.

**Table 13: Security objectives for the TOE rationale**

Security objectives	Rationale
O.ACCOUNTABILITY	<p>[OSPP] FAU_GEN.1 defines the auditable events that must be generated to diagnose the cause of unexpected system behavior.</p> <p>[OSPP] FTP_ITC_EXT.1 provides a mechanism for the TSF to transmit the audit data to a remote system.</p> <p>[BT] FAU_GEN.1/BT supports the objective by requiring the TSF to specify the Bluetooth-related auditable events for which it will generate audit records.</p>
O.INTEGRITY	<p>[OSPP] FPT_SBOP_EXT.1 enforces stack buffer overflow protection that makes it more difficult to exploit running code.</p> <p>[OSPP] FPT_ASLR_EXT.1 prevents attackers from exploiting code that executes in static known memory locations.</p>

Security objectives	Rationale
	<p>[OSPP] <a href="#">FPT_TUD_EXT.1</a> and <a href="#">FPT_TUD_EXT.2</a> enforce integrity of software updates.</p> <p>[OSPP] <a href="#">FCS_COP.1(2)</a>, <a href="#">FCS_COP.1(3)</a>, and <a href="#">FCS_COP.1(4)</a> provide the cryptographic mechanisms that are used to verify integrity values.</p> <p>[OSPP] <a href="#">FPT_ACF_EXT.1</a> guarantees the integrity of critical components by preventing unauthorized modifications of them.</p> <p>[OSPP] <a href="#">FIA_X509_EXT.1</a> provides X.509 certificates as a way of validating software integrity.</p> <p>[OSPP] <a href="#">FPT_TST_EXT.1</a> verifies the integrity of stored code.</p> <p>[OSPP] <a href="#">FPT_W^X_EXT.1</a> prevents execution of data in writable memory.</p> <p>[OSPP] <a href="#">FIA_UAU.5</a> provides mechanisms that prevent untrusted users from accessing the TSF and <a href="#">FIA_AFL.1</a> prevents brute-force authentication attempts.</p> <p>[OSPP] <a href="#">FTP_ITC_EXT.1</a> provides trusted remote communications which makes a remote authenticated session less susceptible to compromise.</p> <p>[BT] <a href="#">FAU_GEN.1/BT</a> supports the objective by requiring the TSF to specify the Bluetooth-related auditable events for which it will generate audit records.</p>
<b>O.MANAGEMENT</b>	<p>[OSPP] <a href="#">FMT_SMF_EXT.1</a> defines the TOE's management functions and <a href="#">FMT_MOF_EXT.1</a> defines the privileges required to invoke them.</p> <p>[OSPP] <a href="#">FTP_TRP.1</a> provides one or more secure remote interfaces for management of the TSF and <a href="#">FTA_TAB.1</a> provides actionable warnings against misuse of these interfaces.</p> <p>[BT] <a href="#">FMT_MOF_EXT.1/BT</a> supports the objective by restricting the ability to perform Bluetooth-related management functions to the Administrator.</p> <p>[BT] <a href="#">FMT_SMF_EXT.1/BT</a> supports the objective by specifying the Bluetooth-related management functions that the TSF must perform.</p>
<b>O.PROTECTED_STORAGE</b>	<p>[OSPP] <a href="#">FCS_STO_EXT.1</a> provides a mechanism by which the TOE can designate data as 'sensitive' and subsequently require it to be encrypted.</p> <p>[OSPP] <a href="#">FCS_COP.1(1)</a> defines the symmetric algorithm used to encrypt and decrypt sensitive data.</p> <p>[OSPP] <a href="#">FCS_RBG_EXT.1</a> defines the random bit generator used to create the symmetric keys used to perform this encryption and decryption.</p> <p>[OSPP] <a href="#">FDP_ACF_EXT.1</a> enforces logical access control on stored data.</p>
<b>O.PROTECTED_COMMS</b>	<p>[OSPP] <a href="#">FCS_TLSC_EXT.1</a>, <a href="#">FCS_TLSC_EXT.2</a>, and <a href="#">FCS_TLSC_EXT.4</a> define the ability of the TOE to act as a TLS client as a method of enforcing protected communications.</p> <p>[OSPP] <a href="#">FCS_CKM.1</a>, <a href="#">FCS_CKM.2</a>, <a href="#">FCS_CKM_EXT.4</a>, <a href="#">FCS_COP.1(1)</a>, <a href="#">FCS_COP.1(2)</a>, <a href="#">FCS_COP.1(3)</a>, <a href="#">FCS_COP.1(4)</a>, and <a href="#">FCS_RBG_EXT.1</a> define the cryptographic operations and key lifecycle activity used to support the establishment of protected communications.</p> <p>[OSPP] <a href="#">FIA_X509_EXT.1</a> defines how the TSF validates x.509 certificates as part of establishing protected communications.</p> <p>[OSPP] <a href="#">FIA_X509_EXT.2</a> defines the trusted communication protocols for which the TOE must perform certificate validation operations.</p>

Security objectives	Rationale
	<p>[OSPP] FTP_ITC_EXT.1 defines the trusted communications channels supported by the TOE.</p> <p>[BT] FCS_CKM_EXT.8 supports the objective by requiring the TSF to specify how ECDH key pairs will be refreshed.</p> <p>[BT] FIA_BLT_EXT.1 supports the objective by ensuring that Bluetooth communications are not initiated without user approval.</p> <p>[BT] FIA_BLT_EXT.2 supports the objective by requiring the TSF to implement Bluetooth mutual authentication.</p> <p>[BT] FIA_BLT_EXT.3 supports the objective by preventing Bluetooth spoofing by rejecting connections with duplicate device addresses.</p> <p>[BT] FIA_BLT_EXT.4 supports the objective by defining the TSF's implementation of Bluetooth Secure Simple Pairing.</p> <p>[BT] FIA_BLT_EXT.6 supports the objective by requiring the TSF to specify the Bluetooth profiles that it requires explicit user authorization to grant access to for trusted devices.</p> <p>[BT] FIA_BLT_EXT.7 supports the objective by requiring the TSF to specify the Bluetooth profiles that it requires explicit user authorization to grant access to for untrusted devices.</p> <p>[BT] FTP_BLT_EXT.1 supports the objective by requiring the TSF to implement encryption to protect Bluetooth communications.</p> <p>[BT] FTP_BLT_EXT.2 supports the objective by requiring the TSF to prevent data transmission over Bluetooth if the paired device is not using encryption.</p> <p>[BT] FTP_BLT_EXT.3/BR support the objective by requiring the TSF to implement a minimum encryption key size for Bluetooth BR/EDR.</p> <p>[BT] FTP_BLT_EXT.3/LE support the objective by requiring the TSF to implement a minimum encryption key size for Bluetooth LE.</p>

## 6.3 Security Assurance Requirements

The security assurance requirements (SARs) for the TOE are defined in the OSPPV4.2.1 protection profile; and defined in CC assurance package.

The following table shows the SARs, and the operations performed on the components according to CC part 3: iteration (Iter.), refinement (Ref.), assignment (Ass.) and selection (Sel.).

Table 14: SARs

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
ALC Life-cycle support	ALC_TSU_EXT.1 Timely Security Updates	OSPPV4.2.1	No	No	No	No
	ALC_CMC.1 Labelling of the TOE	CC	No	No	No	No
	ALC_CMS.1 TOE CM coverage	CC	No	No	No	No
ASE Security Target evaluation	ASE_CCL.1 Conformance claims	CC	No	No	No	No
	ASE_ECD.1 Extended components definition	CC	No	No	No	No
	ASE_INT.1 ST introduction	CC	No	No	No	No

Security assurance class	Security assurance requirement	Source	Operations			
			Iter.	Ref.	Ass.	Sel.
	ASE_OBJ.2 Security objectives	CC	No	No	No	No
	ASE_REQ.2 Derived security requirements	CC	No	No	No	No
	ASE_SPD.1 Security problem definition	CC	No	No	No	No
	ASE_TSS.1 TOE summary specification	CC	No	No	No	No
ADV Development	ADV_FSP.1 Basic functional specification	CC	No	No	No	No
AGD Guidance documents	AGD_OPE.1 Operational user guidance	CC	No	No	No	No
	AGD_PRE.1 Preparative procedures	CC	No	No	No	No
ATE Tests	ATE_IND.1 Independent testing - conformance	CC	No	No	No	No
AVA Vulnerability assessment	AVA_VAN.1 Vulnerability survey	CC	No	No	No	No

## 6.3.1 ALC Life-cycle support

### 6.3.1.1 ALC\_TSU\_EXT.1 Timely Security Updates

Developer action elements:

#### ALC\_TSU\_EXT.1.1D

The developer shall provide a description in the TSS of how timely security updates are made to the OS.

#### ALC\_TSU\_EXT.1.2D

The developer shall provide a description in the TSS of how users are notified when updates change security properties or the configuration of the product.

Content and presentation elements:

#### ALC\_TSU\_EXT.1.1C

The description shall include the process for creating and deploying security updates for the OS software.

#### ALC\_TSU\_EXT.1.2C

The description shall include the mechanisms publicly available for reporting security issues pertaining to the OS.

Evaluator action elements:

#### ALC\_TSU\_EXT.1.1E

The evaluator will confirm that the information provided meets all requirements for content and presentation of evidence.

## 6.4 Security Assurance Requirements Rationale

SAR rationales are provided by the PPs to which this ST conforms. [Section 2](#) contains the list of PPs.



## 7 TOE Summary Specification

### 7.1 TSS Security Assurance Evaluation Activity

#### 7.1.1 Timely security updates (ALC\_TSU\_EXT.1)

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

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

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

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

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

### 7.2 TOE Security Functionality

#### 7.2.1 Audit

##### 7.2.1.1 FAU\_GEN.1 Audit data generation

**PP Origin:** OSPP

**SFR Link:** [FAU\\_GEN.1](#)

Audit events are generated for the following audit functions:

- Start-up and shut-down of the 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 following information:

- Date and time
- Type of event
- Subject identity (if applicable)
- Outcome (success or failure)

The logging system captures messages across all levels of the system, and it stores the log data in memory and data store on disk. Audit events are only accessible by administrators. Audit events can be viewed via the command line. If the username is longer than 8 characters, the system uses the user ID to display. To access the audit logs, the TOE provides built-in utilities "audit", "praudit", and "auditreduce". All audit records are Basic Security Module (BSM) compliant, and any BSM Audit Tool could be used for viewing audit logs.

### 7.2.1.2 FAU\_GEN.1/BT Audit data generation (Bluetooth)

**PP Origin:** *BT*

**SFR Link:** [FAU\\_GEN.1/BT](#)

The TOE generates Bluetooth audit records after the "Bluetooth for macOS" configuration profile is installed on the TOE. This profile is obtainable from the Apple Developer website under "Profiles and Logs » macOS" along with instructions. The administrator can use the macOS *sysdiagnose* feature to obtain these records in the form of log files. The */private/var/tmp* folder will contain the log files.

The TOE audits the following events:

- Start-up and shutdown of the audit functions
- Specifically defined auditable events listed in [Table 9](#)

Each audit record contains the following information:

- Date and time
- Type of event
- Subject identity
- Outcome (success or failure)
- Additional information as specified in [Table 9](#)

The TOE does not generate audit records for Bluetooth duplicate connection attempts ([FIA\\_BLT\\_EXT.3](#)) because rejections happen at the Host Controller Interface (HCI) layer.

## 7.2.2 Cryptography

The security features that use cryptography in this ST are the following:

- Bluetooth
- Keychains
- Secure boot
- TLS client
- Trusted update

The cryptographic modules used to implement the above security features are the following:

- Apple corecrypto Module v13.0 [Apple ARM, User, Software, SL1] (user space)
- Apple corecrypto Module v13.0 [Apple ARM, Kernel, Software, SL1] (kernel space)
- Apple corecrypto Module v13.0 [Apple silicon, Secure Key Store, Hardware, SL2] (SKS)
- Apple corecrypto Module v13.0 [Intel, User, Software, SL1] (user space)
- Apple corecrypto Module v13.0 [Intel, Kernel, Software, SL1] (kernel space)
- Apple corecrypto Module v13.0 [Apple ARM, Secure Key Store, Hardware, SL2] (SKS)
- Bluetooth hardware

Bluetooth uses the user space module for all cryptographic operations except for the Bluetooth AES-CCM-128 bulk encryption which is implemented by the Bluetooth hardware.

Keychains use the user space module with the root key anchored by the SKS module.

Secure boot uses the kernel space module and SKS module.

TLS and trusted update use the user space module for all cryptographic operations.

Table 15 lists the algorithms discussed in the following subsections.

**Table 15: Cryptographic algorithm table**

SFR	Algorithm	Capabilities	Usage
FCS_CKM.1	RSA KeyGen	2048, 3072, 4096	Client authentication in TLS client with mutual authentication
	ECDSA KeyGen	P-256, P-384, P-521	Bluetooth SSP (P-256) Client authentication in TLS client with mutual authentication and key establishment using ephemeral keys in TLS client (ECDHE)
FCS_CKM.2	RSA RSAES-PKCS1-v1_5 Key Establishment	2048, 3072, 4096	Key establishment in TLS client
	ECC Key Establishment (KAS-ECC)	P-256, P-384, P-521	Bluetooth SSP (P-256) Key establishment in TLS client
FCS_COP.1(1)	AES-CBC, AES-GCM	128-bit, 256-bit	Keychains (AES-GCM-256) TLS client (AES-CBC, AES-GCM)
	AES-CCM	128-bit	Bluetooth SSP (AES-CCM-128)
FCS_COP.1(2)	SHA-1, SHA-256, SHA-384, SHA-512		Secure boot (SHA-256 in Apple silicon and SHA-512 in "Intel with T2") TLS client Trusted update (SHA-256 in Apple silicon and SHA-512 in "Intel with T2")
FCS_COP.1(3)	RSA SigVer	2048, 3072, 4096 with: SHA-1, SHA-256, SHA-384, SHA-512	Secure boot (4096 bits with SHA-256 in "Intel with T2") TLS client Trusted update (4096 bits with SHA-256 in "Intel with T2")
	ECDSA SigVer	P-256, P-384, P-521 with: SHA-1, SHA-256, SHA-384, SHA-512	Secure boot (P-512 with SHA-512 in Apple silicon) TLS client Trusted update (P-512 with SHA-512 in Apple silicon)
FCS_COP.1(4)	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512		Bluetooth BR/EDR (HMAC-SHA-256) TLS client
FCS_RBG_EXT.1	CTR_DRBG	AES	Bluetooth TLS client

### 7.2.2.1 FCS\_CKM.1 Cryptographic key generation

**PP Origin:** OSPP

**SFR Link:** [FCS\\_CKM.1](#)

The TOE supports generation of 2048-bit, 3072-bit, and 4096-bit RSA keys conforming to FIPS PUB 186-4 "Digital Signature Standard (DSS)", Appendix B.3. The TOE provides RSA key generation for being used in TLS sessions with client authentication.

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

Please refer to [Table 15](#) for details.

### 7.2.2.2 FCS\_CKM.2 Cryptographic key establishment

**PP Origin:** OSPP

**SFR Link:** [FCS\\_CKM.2](#)

The TOE supports cryptographic key establishment using the following schemes:

- RSAES-PKCS1-v1\_5 RSA-based key establishment with 2048-bit, 3072-bit, and 4096-bit keys as specified in Section 7.2 of RFC 8017.
- Elliptic curve-based key establishment with NIST curves P-256, P-384, and P-521 as specified in NIST SP 800-56A rev 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".

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

Elliptic curve-based key establishment is used in TLS sessions when ciphersuites with ECDHE key exchange are negotiated. Bluetooth SSP initialization also uses elliptic curve-based key establishment.

Please refer to [Table 15](#) for details.

### 7.2.2.3 FCS\_CKM\_EXT.4 Cryptographic key destruction

**PP Origin:** OSPP

**SFR Link:** [FCS\\_CKM\\_EXT.4](#)

The TOE includes a Keychain Access program that allows users the ability to add, remove, and manage certificates and private keys. Please see [Section 7.2.2.10](#) below for Keychain details. The metadata key is a Data Encryption Key (DEK) stored in non-volatile memory. Each Keychain item is protected with an individual DEK. The metadata of all Keychain items is collectively encrypted with another DEK.

Each DEK is wrapped by the Secure Enclave with one 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 passcode.

All DEKs and KEKs are always stored in wrapped form in non-volatile store. 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).

The wrapped keys held in non-volatile store are cryptographically destroyed by destroying the KEK wrapping key.

The plaintext keys held in volatile store are overwritten with zeros.

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

#### 7.2.2.4 FCS\_CKM\_EXT.8 Bluetooth key generation

**PP Origin:** *BT*

**SFR Link:** *FCS\_CKM\_EXT.8*

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

#### 7.2.2.5 FCS\_COP.1(1) Cryptographic operation - encryption/decryption

**PP Origin:** *OSPP*

**SFR Link:** *FCS\_COP.1(1)*

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

- CBC as specified in NIST SP 800-38A (for TLS)
- GCM as specified in NIST SP 800-38D (for TLS, Keychain)
- CCM (128-bit) as specified in NIST SP 800-38C (for Bluetooth)

Please refer to [Table 15](#) for details.

#### 7.2.2.6 FCS\_COP.1(2) Cryptographic operation - hashing

**PP Origin:** *OSPP*

**SFR Link:** *FCS\_COP.1(2)*

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

The TOE supports the following hash algorithms: SHA-1, SHA-256, SHA-384, and SHA-512. The message digest sizes supported are: 160 bits, 256 bits, 384 bits, and 512 bits.

Please refer to [Table 15](#) for details.

#### 7.2.2.7 FCS\_COP.1(3) Cryptographic operation - signing

**PP Origin:** *OSPP*

**SFR Link:** *FCS\_COP.1(3)*

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

- RSA digital signature algorithm conforming to FIPS Pub 186-4, "Digital Signature Standard (DSS)", Section 4. The RSA key sizes supported are: 2048 bits, 3072 bits, and 4096-bit.
- Elliptical curve digital signature algorithm conforming to FIPS Pub 186-4, "Digital Signature Standard (DSS)", Section 5. The TOE supports curves P-256, P-384, and P-521.

Please refer to [Table 15](#) for details.

#### 7.2.2.8 FCS\_COP.1(4) Cryptographic operation - keyed-hash message authentication

**PP Origin:** *OSPP*

**SFR Link:** *FCS\_COP.1(4)*

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

- HMAC-SHA-1

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

The TOE supports key sizes 8 through 262144 bits for all HMAC algorithms.

Please refer to [Table 15](#) for details.

### 7.2.2.9 FCS\_RBG\_EXT.1 Random bit generation

**PP Origin:** OSPP

**SFR Link:** [FCS\\_RBG\\_EXT.1](#)

The TOE uses a CTR\_DRBG(AES) to generate random bits. The DRBG is seeded by an entropy source that accumulates entropy from a software-based noise (interrupts) source with a minimum of 256 bits of entropy.

Please refer to [Table 15](#) for details.

### 7.2.2.10 FCS\_STO\_EXT.1 Storage of sensitive data

**PP Origin:** OSPP

**SFR Link:** [FCS\\_STO\\_EXT.1](#)

The TOE stores the following sensitive data:

- TOE usernames used for authentication are maintained by the local directory services.
- Trusted Certificates are used for establishing TLS sessions and are stored in the macOS Keychain.
- Private Keys are used for establishing TLS session and are stored in the macOS Keychain.

The TOE does not store login passwords. Instead, the TOE converts the user's typed password into a key used to unwrap the user's class keys. If the typed password is correct (i.e., it successfully unwraps the user's class keys), the user is considered authenticated; otherwise, authentication fails.

TOE usernames are handled via the "Users & Group" GUI described in section 3.2 of the [\[CCGUIDE\]](#).

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

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.

Keychain items are encrypted using two different AES-GCM-256 keys:

- Table key (metadata key)
- Per-row key (secret key)

Keychain metadata (all attributes other than `kSecValue`) is encrypted with the metadata key to speed searches while the secret value (`kSecValueData`) is encrypted with the secret key. The metadata key is protected by the Secure Enclave (platform-provided HW storage) but is cached in the application processor to allow fast queries of the keychain.

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

### 7.2.2.11 FCS\_TLSC\_EXT.1 TLS client protocol

**PP Origin:** OSPP

**SFR Link:** [FCS\\_TLSC\\_EXT.1](#)

The TOE implements TLS 1.2 (RFC 5246) client functionality supporting the following cipher suites:

- TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA as defined in RFC 5246
- TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA as defined in RFC 5246
- TLS\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5288
- TLS\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5288
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256 as defined in RFC 5289
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384 as defined in RFC 5289

The TOE also supports the following TLS 1.2 cipher suites not specified in [OSPPv4.2.1][4](#):

- TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_CBC\_SHA
- TLS\_ECDHE\_ECDSA\_WITH\_AES\_256\_CBC\_SHA
- TLS\_ECDHE\_RSA\_WITH\_AES\_128\_CBC\_SHA
- TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA

The TOE verifies that server certificate is valid according to [FIA\\_X509\\_EXT.1](#) and that the presented identifier matches the reference identifier according to RFC 6125. The reference identifiers supported are DNS and IP addresses in the SAN. Wildcards are supported. The TOE does not support certificate pinning. The TOE does not establish a trusted channel if the server certificate is invalid.

### 7.2.2.12 FCS\_TLSC\_EXT.2 TLS client protocol

**PP Origin:** OSPP

**SFR Link:** [FCS\\_TLSC\\_EXT.2](#)

The TOE, by default, presents the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: secp256r1, secp384r1, and secp521r1.

### 7.2.2.13 FCS\_TLSC\_EXT.4 TLS client protocol

**PP Origin:** OSPP

**SFR Link:** [FCS\\_TLSC\\_EXT.4](#)

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

## 7.2.3 User data protection

### 7.2.3.1 FDP\_ACF\_EXT.1 Access controls for protecting user data

**PP Origin:** OSPP

**SFR Link:** [FDP\\_ACF\\_EXT.1](#)

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

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

1. If the app's sandbox forbids the requested access, the request is denied.
2. If ownership checking has been disabled for the volume in question by the system administrator (with a checkbox in its Finder Get Info window), the request is granted.
3. If an access control entry exists on the file, it is evaluated and used to determine access rights.
4. If a file flag prohibits the operation, the operation is denied.
5. Otherwise, if the user ID matches the owner of the file, the "user" permissions (also called "owner" permissions) are used.
6. Otherwise, if the group ID matches the group for the file, the "group" permissions are used.
7. 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.

### **POSIX ACLs**

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

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

- Inherited (this ACE was inherited)
- File Inherit (this ACE should be inherited by files created within this directory)
- Directory Inherit (this ACE should be inherited by directories created within this directory)
- Inherit Only (this ACE should not be checked during authorization)
- No Propagate Inherit (this ACE should be inherited only by direct children; that is, the ACE should lose any Directory Inherit or File Inherit bit when inherited)

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

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

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

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

1. Explicitly specified deny associations



## 2. Explicitly specified allow associations

Inherited associations appear in the same order in which they appeared in the parent. Since ACEs can be inherited, administrators can control the fine-grained permissions of files created in a directory by assigning inheritable ACEs to the directory. Doing so saves the work of assigning ACEs to each file individually. In addition, because ACEs can apply to groups of users, administrators can assign permissions to groups rather than having to specify permissions for each individual. Applying access security to directories and groups rather than to files and individuals saves administrator time and gives better file system performance in many circumstances.

### Unix Permissions

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

- UID, short for User ID. Commonly referred to as the File's Owner.
- GID, short for Group ID.
- Flags that include permission bits and other related attributes.

The flags for a file or directory are a 16-bit value that is often represented as a three-digit or four-digit octal value (with the top four or seven bits dropped). The Owner, Group, and Other bit sets contain three bits: read, write, execute (rwx for short).

### BSD File Flags

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

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

## 7.2.4 Identification and authentication

### 7.2.4.1 FIA\_AFL.1 Authentication failure handling

**PP Origin:** *OSPP*

**SFR Link:** [FIA\\_AFL.1](#)

The TOE will detect when an administrator-configurable integer within 1-50 unsuccessful authentication attempts for authentication based on username and password attempts have been met. Once the specified number of unsuccessful authentication attempts for an account has been met, the TOE will lock out the account.

### 7.2.4.2 FIA\_BLT\_EXT.1 Bluetooth user authorization

**PP Origin:** *BT*

**SFR Link:** [FIA\\_BLT\\_EXT.1](#)

The TOE supports SSP and the following Bluetooth association models:

- Numeric comparison
- Passkey entry

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

Manual authorization is implicitly configured since pairing can only occur when explicitly authorized through the [System Settings » Bluetooth](#) interface of the TOE device. During the pairing time, another device (or the TOE) can send a pairing request. Commonly, a six-digit number is displayed on both sides, which must be manually

matched by a user (i.e., the PIN is shown and the user must accept it before the pairing completes). If one device does not support this automatic exchange of a PIN, a window for entering a manual PIN is shown. That PIN must match on both sides.

This applies to applications that use Bluetooth also.

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

### 7.2.4.3 FIA\_BLT\_EXT.2 Bluetooth mutual authentication

**PP Origin:** *BT*

**SFR Link:** *FIA\_BLT\_EXT.2*

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

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

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

### 7.2.4.4 FIA\_BLT\_EXT.3 Rejection of duplicate Bluetooth connections

**PP Origin:** *BT*

**SFR Link:** *FIA\_BLT\_EXT.3*

Bluetooth devices may not establish more than one connection. Multiple connection attempts (i.e., pairing and session initialization attempts) from the same BD\_ADDR for an established connection will be discarded. For details of the security of Bluetooth/LE see the Bluetooth Specifications [BT\_SPEC].

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

### 7.2.4.5 FIA\_BLT\_EXT.4 Secure Simple Pairing

**PP Origin:** *BT*

**SFR Link:** *FIA\_BLT\_EXT.4*

Devices that want to pair with the TOE via Bluetooth are required by the TOE to use Secure Simple Pairing, which uses ECDH-based authentication and key exchange and AES for data encryption. See the Bluetooth Specifications [BT\_SPEC] for details.

### 7.2.4.6 FIA\_BLT\_EXT.6 Trusted Bluetooth device user authorization

**PP Origin:** *BT*

**SFR Link:** *FIA\_BLT\_EXT.6*

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 1.6)
- Phone Book Access Profile (PBAP)
- Advanced Audio Distribution Profile (A2DP)
- Audio/Video Remote Control Profile (AVRCP 1.4)
- Personal Area Network Profile (PAN)

- Human Interface Device Profile (HID)
- Message Access Profile (MAP)

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

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

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

#### **7.2.4.7 FIA\_BLT\_EXT.7 Untrusted Bluetooth device user authorization**

**PP Origin:** *BT*

**SFR Link:** [FIA\\_BLT\\_EXT.7](#)

See the [TSS for FIA\\_BLT\\_EXT.6](#).

#### **7.2.4.8 FIA\_UAU.5 Multiple authentication mechanisms**

**PP Origin:** *OSPP*

**SFR Link:** [FIA\\_UAU.5](#)

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

For password-based authentication, the user account supports a username and a password. To authenticate, a Password-Based Derivation Key Function 2 (PBKDF2) with SHA-256 is used to generate the user's keybag key from the user's password. If the output key from the PBKDF2 function can unwrap the user's keybag, then the user is granted access; otherwise, the user is denied access.

For smart card authentication, the user initially logs in providing a valid username and password. Once successfully authenticated, a smart card is paired to the user account by connecting the smart card and entering the smart card's PIN to unlock the card. Entering the PIN releases the smart card's certificate (which contains its public key), so it can be stored by the TOE. The certificate is associated with the user's account and the card is considered to be paired with the user. When a user inserts a smart card to authenticate, the user enters the associated PIN to unlock the card. Once unlocked, a signing operation is performed by the card. The TOE verifies the signature using the paired certificate for authentication.

#### **7.2.4.9 FIA\_X509\_EXT.1 X.509 Certificate validation**

**PP Origin:** *OSPP*

**SFR Link:** [FIA\\_X509\\_EXT.1](#)

When an X.509 certificate is presented, the TOE verifies the certificate path and certification validation process by verifying 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 caSigning purpose in the key usage field
- The OS shall validate the revocation status of the certificate using OCSP. The certificate is accepted if its revocation status cannot be determined.

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

- 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 ECU 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 ECU 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 ECU 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 ECU field.

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

#### **7.2.4.10 FIA\_X509\_EXT.2 X.509 Certificate authentication**

**PP Origin:** *OSPP*

**SFR Link:** *FIA\_X509\_EXT.2*

The TOE uses X.509v3 certificates for performing mutual authentication for TLS and HTTPS connections.

### **7.2.5 Security management**

#### **7.2.5.1 FMT\_MOF\_EXT.1 Management of security functions behavior**

**PP Origin:** *OSPP*

**SFR Link:** *FMT\_MOF\_EXT.1*

See the TSS for [FMT\\_SMF\\_EXT.1](#).

#### **7.2.5.2 FMT\_MOF\_EXT.1/BT Management of security functions behavior**

**PP Origin:** *BT*

**SFR Link:** *FMT\_MOF\_EXT.1/BT*

See the TSS for [FMT\\_SMF\\_EXT.1/BT](#).

#### **7.2.5.3 FMT\_SMF\_EXT.1 Specification of management functions**

**PP Origin:** *OSPP*

**SFR Link:** *FMT\_SMF\_EXT.1*

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

The Administrator has access to the following management functions:

- Enable/disable screen lock ("Require password after screen saver begins or display is turned off")
- Configure screen lock inactivity timeout ("Start Screen Saver when inactive")
- Configure local audit storage capacity
- Configure minimum password Length
- Configure minimum number of special characters in password
- Configure host-based firewall
- Configure name/address of remote management server from which to receive management settings
- Configure name/address of the logging server (syslog) to which to send logging records
- Configure audit rules
- Configure name/address of network time server
- Enable/disable automatic software update
- Configure Wi-Fi interface

The User has access to the following management functions:

- Enable/disable screen lock ("Require password after screen saver begins or display is turned off")
- Configure Wi-Fi interface (if not restricted by an Administrator)
- Enable/disable Bluetooth interface

#### **7.2.5.4 FMT\_SMF\_EXT.1/BT Specification of management functions**

**PP Origin:** *BT*

**SFR Link:** *FMT\_SMF\_EXT.1/BT*

The TOE supports both Bluetooth BR/EDR and LE, uses Secure Simple Pairing (SSP) for security, and supports the following Bluetooth profiles:

- Hands-Free Profile (HFP 1.6)
- Phone Book Access Profile (PBAP)
- Advanced Audio Distribution Profile (A2DP)
- Audio/Video Remote Control Profile (AVRCP 1.4)
- Personal Area Network Profile (PAN)
- Human Interface Device Profile (HID)
- Message Access Profile (MAP)

Different TOE devices support different versions of Bluetooth. See [Table 16](#) for the mapping of TOE devices to Bluetooth versions.

##### BT-1

The TOE allows an administrator and user to enable and disable the Bluetooth Discoverable (for BR/EDR) and Advertising (for LE) modes.

## 7.2.6 Protection of the TSF

### 7.2.6.1 FPT\_ACF\_EXT.1 Access controls

**PP Origin:** OSPP

**SFR Link:** [FPT\\_ACF\\_EXT.1](#)

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

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

- Kernel drivers and modules
  - /System/Library/Extensions/
- Shared libraries
  - Library/Frameworks
  - /System/Library/Frameworks
  - /System/Library/PrivateFrameworks
- Applications
  - /System/Applications
    - > Apps that are installed with the TOE and updated as part of TOE updates
  - /Applications
    - > Apps that are installed with the TOE but updated independently

Standard file permissions (FDP\_ACF\_EXT.1) are used to protect the following from unauthorized modification:

- Security audit logs
  - /var/audit
- System configuration files
  - /Library/Preferences - system-wide "preferences"
  - ~/Library/Preferences - User-specific "preferences." A user has permission to modify their own configuration files.
  - /etc/security/audit-control
- TSF-data
  - /var
  - /var/db/mds/messages/<PID>/se\_SecurityMessages – each se\_SecurityMessages file contains messages for the user represented by <PID>. A user has permission to modify their own data.
  - /var/folders/zz
  - /var/folders/<non-zz directories> - user-specific TSF data. A user has permission to modify their own data.
- System-wide credentials repositories (accessed through local directory services)
  - /private/var/db/dslocal/nodes/Default/
- Applications
  - /Applications

- Apps installed by a traditional installer
- Apps copied into /Applications

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

### 7.2.6.2 FPT\_AS LR\_EXT.1 Address space layout randomization (ASLR)

**PP Origin:** OSPP

**SFR Link:** [FPT\\_AS LR\\_EXT.1](#)

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

### 7.2.6.3 FPT\_SBOP\_EXT.1 Stack buffer overflow protection

**PP Origin:** OSPP

**SFR Link:** [FPT\\_SBOP\\_EXT.1](#)

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

- ASLR to randomize executable 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.

### 7.2.6.4 FPT\_TST\_EXT.1 Boot integrity

**PP Origin:** OSPP

**SFR Link:** [FPT\\_TST\\_EXT.1](#)

The boot process for an Apple silicon Mac is as follows:

1. The application 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 an Intel-based Mac with an Apple T2 security chip is as follows:

- T2
  1. The T2 loads the Boot ROM.
  2. The Boot ROM validates the iBoot signature using the Apple Root CA public key.
  3. iBoot validates the T2 kernel cache signature.
  4. The T2 kernel cache evaluates the UEFI firmware signature.
  5. The UEFI firmware begins the boot process on the Intel CPU.
- Intel CPU
  1. The UEFI firmware validates the boot.efi signature.
  2. boot.efi validates the macOS immutable kernel signature.
  3. The TOE (macOS) begins execution.

For both hardware platforms, 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.

### 7.2.6.5 FPT\_TUD\_EXT.1 Trusted update

**PP Origin:** OSPP

**SFR Link:** [FPT\\_TUD\\_EXT.1](#)

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

Signature verification of the TOE image is performed by the SEP. As the kernel boots, each stage of the boot process validates the signatures of the next stage.

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

#### **Apple silicon**

*Algorithm:* ECDSA P-521 sigver

*Standard:* FIPS PUB 186-4

*Modules:*

- Apple corecrypto Module v13.0 [Apple ARM, User, Software, SL1]
- Apple corecrypto Module v13.0 [Apple ARM, Kernel, Software, SL1]
- Apple corecrypto Module v13.0 [Apple silicon, Secure Key Store, Hardware, SL2]

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

#### **Intel with T2**

*Algorithm:* RSA 4096 sigver

*Standard:* IEEE 1619

*Modules:*

- Apple corecrypto Module v13.0 [Intel, User, Software, SL1]
- Apple corecrypto Module v13.0 [Intel, Kernel, Software, SL1]
- Apple corecrypto Module v13.0 [Apple ARM, Secure Key Store, Hardware, SL2]



On "Intel with T2" Macs, signatures are verified using RSA 4096-bit and SHA-256. The CA public key is embedded in the SEP's Boot ROM code during manufacturing. The TOE image is signed using this public key's corresponding private key.

### 7.2.6.6 FPT\_TUD\_EXT.2 Trusted update for application software

**PP Origin:** OSPP

**SFR Link:** [FPT\\_TUD\\_EXT.2](#)

See the TSS for [FPT\\_TUD\\_EXT.1](#).

### 7.2.6.7 FPT\_W^X\_EXT.1 Write XOR execute memory pages

**PP Origin:** OSPP

**SFR Link:** [FPT\\_W^X\\_EXT.1](#)

#### **Apple silicon**

As a general rule, application memory has either write or execute permission but not both simultaneously. There exists a special condition where memory pages marked as both writable and executable can be used only by apps under tightly controlled conditions: the kernel checks for the presence of the Apple-only dynamic code-signing entitlement. Even then, only a single mmap call can be made to request a writable and executable page, which is given a randomized address.

The Safari web browser, Perl language, Python language, and JavaScript language contain components that use Just-In-Time (JIT) code compilation technology and require both write and execute permission to memory.

#### **Intel with T2**

As a general rule, application memory has either write or execute permission but not both simultaneously when the corresponding binary executable contains the hardened execution flag. This flag is consulted by the application loading mechanism of the operating system which prevents the concurrent permissions of write and execute on current and newly allocated memory.

## 7.2.7 TOE access

### 7.2.7.1 FTA\_TAB.1 Default TOE access banners

**PP Origin:** OSPP

**SFR Link:** [FTA\\_TAB.1](#)

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

## 7.2.8 Trusted path/channels

### 7.2.8.1 FTP\_BLT\_EXT.1 Bluetooth encryption

**PP Origin:** BT

**SFR Link:** [FTP\\_BLT\\_EXT.1](#)

The TOE supports Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR) and Low Energy (LE). Bluetooth 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-based authentication and key exchange and AES for data encryption. This applies to both BR/EDR and LE. Bluetooth encryption is always enabled.

### 7.2.8.2 FTP\_BLT\_EXT.2 Persistence of Bluetooth encryption

**PP Origin:** *BT*

**SFR Link:** [FTP\\_BLT\\_EXT.2](#)

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

### 7.2.8.3 FTP\_BLT\_EXT.3/BR Bluetooth encryption parameters (BR/EDR)

**PP Origin:** *BT*

**SFR Link:** [FTP\\_BLT\\_EXT.3/BR](#)

Connections via BR/EDR and LE are secured using 128-bit AES Counter with CBC-MAC (AES-CCM-128) mode. No other key sizes are supported; thus, smaller key sizes cannot be negotiated.

### 7.2.8.4 FTP\_BLT\_EXT.3/LE Bluetooth encryption parameters (LE)

**PP Origin:** *BT*

**SFR Link:** [FTP\\_BLT\\_EXT.3/LE](#)

See the TSS for [FTP\\_BLT\\_EXT.3/BR](#) for LE connections.

### 7.2.8.5 FTP\_ITC\_EXT.1 Trusted channel communication

**PP Origin:** *OSPP*

**SFR Link:** [FTP\\_ITC\\_EXT.1](#)

The TOE uses TLS as conforming to [FCS\\_TLSC\\_EXT.1](#) to provide a trusted channel between itself and authorized IT entities. The update server and other authenticated TLS servers are authorized IT entities.

### 7.2.8.6 FTP\_TRP.1 Trusted path

**PP Origin:** *OSPP*

**SFR Link:** [FTP\\_TRP.1](#)

The TOE provides a trusted path between itself and local users that provides assured identification of its endpoints. The trusted path is initiated by the local user. The TOE does not support a remote administrative communication path in the evaluated configuration.

# 8 Abbreviations, Terminology, and References

## 8.1 Abbreviations

<b>ABM</b>	Apple Business Manager
<b>ACE</b>	Access Control Entry
<b>AES</b>	Advanced Encryption Standard
<b>app</b>	Application
<b>APFS</b>	Apple File System
<b>API</b>	Application Programming Interface
<b>ASLR</b>	Address Space Layout Randomization
<b>BD_ADDR</b>	Bluetooth Device Address
<b>BR/EDR</b>	Basic Rate/Enhanced Data Rate
<b>BSD</b>	Berkeley Software Distribution
<b>BSM</b>	Basic Security Module
<b>CA</b>	Certificate Authority
<b>CBC</b>	Cypher Block Chaining
<b>CC</b>	Common Criteria
<b>CCM</b>	Counter with CBC-MAC
<b>CEM</b>	Common Evaluation Methodology
<b>CIFS</b>	Common Internet File System
<b>CMC</b>	Certificate Management over CMS

<b>CMS</b>	Cryptographic Message Syntax
<b>CSP</b>	Critical Security Parameters
<b>CVE</b>	Common Vulnerabilities and Exposures
<b>DAR</b>	Data At Rest
<b>DEK</b>	Data Encryption Key
<b>DEP</b>	Data Execution Prevention
<b>DNS</b>	Domain Name System
<b>DRBG</b>	Deterministic Random Bit Generator
<b>DSS</b>	Digital Signature Standard
<b>ECC</b>	Elliptic Curve Cryptography
<b>ECDH</b>	Elliptic Curve Diffie-Hellman
<b>ECDHE</b>	ECDH Ephemeral
<b>EKU</b>	extendedKeyUsage
<b>EST</b>	Enrollment over Secure Transport
<b>GCM</b>	Galois/Counter Mode
<b>GID</b>	Group Identifier
<b>GPOS</b>	General Purpose Operating System
<b>HCI</b>	Host Controller Interface
<b>HMAC</b>	Keyed-hash Message Authentication Code
<b>HTTPS</b>	Hypertext Transfer Protocol Secure

<b>ID</b>	Identifier or Identity
<b>IP</b>	Internet Protocol
<b>KAS</b>	Key Agreement Scheme
<b>KEK</b>	Key Encryption Key
<b>L2CAP</b>	Logical Link Control and Adaptation Protocol
<b>LE</b>	Low Energy
<b>LLB</b>	Low-Level Bootloader
<b>MAC</b>	Message Authentication Code
<b>OCSP</b>	Online Certificate Status Protocol
<b>OID</b>	Object Identifier
<b>OS</b>	Operating System
<b>PBKDF</b>	Password-Based Key Derivation Function
<b>PGP</b>	Pretty Good Privacy
<b>PII</b>	Personally Identifiable Information
<b>PIN</b>	Personal Identification Number
<b>PKI</b>	Public Key Infrastructure
<b>POSIX</b>	Portable Operating System Interface
<b>PP</b>	Protection Profile
<b>RA</b>	Registration Authority
<b>RBG</b>	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

**SEP**

Secure Enclave Processor

**SFR**

Security Functional Requirement

**SHA**

Secure Hash Algorithm

**SL1**

Security Level 1 (FIPS 140-3)

**SMB**

Server Message Block

**SoC**

System on a Chip

**SSP**

Secure Simple Pairing

**ST**

Security Target

**TLS**

Transport Layer Security

**TOE**

Target of Evaluation

**TRNG**

True Random Number Generator

**TSF**

TOE Security Functionality

**TSFI**

TSF Interface

**TSS**

TOE Summary Specification

**UDID**

Unique Device Identifier

**UID**

User Identifier

**UUID**

Universally Unique Identifier

**XNU**

X is Not Unix

## 8.2 Terminology

This section contains definitions of technical terms that are used with a meaning specific to this document. Terms defined in the [CC] are not reiterated here, unless stated otherwise.

**Administrator**

An administrator is responsible for management activities, including setting policies that are applied by the enterprise on the operating system. This administrator could be acting remotely through a management server, from which the system receives configuration policies. An administrator can enforce settings on the system which cannot be overridden by non-administrator users.

**API**

A specification of routines, data structures, object classes, and variables that allows an application to make use of services provided by another software component, such as a library. APIs are often provided for a set of libraries included with the platform.

**app**

Software that runs on a platform and performs tasks on behalf of the user or owner of the platform, as well as its supporting documentation.

**ASLR**

An anti-exploitation feature which loads memory mappings into unpredictable locations. ASLR makes it more difficult for an attacker to redirect control to code that they have introduced into the address space of a process.

**CC**

Common Criteria for Information Technology Security Evaluation.

**CEM**

Common Evaluation Methodology for Information Technology Security Evaluation.

**Credential**

Data that establishes the identity of a user, e.g. a cryptographic key or password.

**CSP**

Information that is either user or system defined and is used to operate a cryptographic module in processing encryption functions including cryptographic keys and authentication data, such as passwords, the disclosure or modification of which can compromise the security of a cryptographic module or the security of the information protected by the module.

**DAR Protection**

Countermeasures that prevent attackers, even those with physical access, from extracting data from non-volatile storage. Common techniques include data encryption and wiping.

**DEP**

An anti-exploitation feature of modern operating systems executing on modern computer hardware, which enforces a non-execute permission on pages of memory. DEP prevents pages of memory from containing both data and instructions, which makes it more difficult for an attacker to introduce and execute code.

**Developer**

An entity that writes OS software. For the purposes of this document, vendors and developers are the same.

**General Purpose Operating System**

A class of OSEs designed to support a wide-variety of workloads consisting of many concurrent applications or services. Typical characteristics for OSEs in this class include support for third-party applications, support for multiple users, and security separation between users and their respective resources. General Purpose Operating Systems also lack the real-time constraint that defines Real Time Operating Systems (RTOS). RTOSes typically power routers, switches, and embedded devices.

**Host-based Firewall**

A software-based firewall implementation running on the OS for filtering inbound and outbound network traffic to and from processes running on the OS.

**OS**

Software that manages physical and logical resources and provides services for applications. The terms *TOE* and *OS* are interchangeable in this document.

**PII**

Any information about an individual maintained by an agency, including, but not limited to, education, financial transactions, medical history, and criminal or employment history and information which can be used to distinguish or trace an individual's identity, such as their name, social security number, date and place of birth, mother's maiden name, biometric records, etc., including any other personal information which is linked or linkable to an individual.

**PP**

An implementation-independent set of security requirements for a category of products.

**SAR**

A requirement to assure the security of the TOE.

**Sensitive Data**

Sensitive data may include all user or enterprise data or may be specific application data such as PII, emails, messaging, documents, calendar items, and contacts. Sensitive data must minimally include credentials and keys. Sensitive data shall be identified in the OS's TSS by the ST author.

**SFR**

A requirement for security enforcement by the TOE.

**ST**

A set of implementation-dependent security requirements for a specific product.

**TOE**

The product under evaluation. In this case, the Operating System and its supporting documentation.

**TSF**

The security functionality of the product under evaluation.

**TSS**

A description of how a TOE satisfies the SFRs in a ST.



**User**

A user is subject to configuration policies applied to the operating system by administrators. On some systems under certain configurations, a normal user can temporarily elevate privileges to that of an administrator. At that time, such a user should be considered an administrator.

**8.3 References**

BT	<b>PP-Module for Bluetooth</b> Version 1.0 Date 2021-04-15 Location <a href="https://www.niap-ccevs.org/Profile/Info.cfm?PPID=425&amp;id=425">https://www.niap-ccevs.org/Profile/Info.cfm?PPID=425&amp;id=425</a>
BT_SPEC	<b>Bluetooth Specifications</b> Author(s) Bluetooth SIG, Inc. Date 2021-07-13 Location <a href="https://www.bluetooth.com/specifications/">https://www.bluetooth.com/specifications/</a>
CC	<b>Common Criteria for Information Technology Security Evaluation</b> Version 3.1R5 Date April 2017 Location <a href="http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf">http://www.commoncriteriaportal.org/files/ccfiles/CCPART1V3.1R5.pdf</a> Location <a href="http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf">http://www.commoncriteriaportal.org/files/ccfiles/CCPART2V3.1R5.pdf</a> Location <a href="http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf">http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R5.pdf</a>
CCGUIDE	<b>Apple macOS 13 Ventura Common Criteria Configuration Guide</b> Version 1.1 Date 2024-01-12 Location <a href="https://www.niap-ccevs.org/MMO/Product/st_vid11347-agd.pdf">https://www.niap-ccevs.org/MMO/Product/st_vid11347-agd.pdf</a>
CFG_GPOS-BT_V1.0	<b>PP-Configuration for General Purpose Operating Systems and Bluetooth</b> Version 1.0 Date 2021-04-15 Location <a href="https://www.niap-ccevs.org/MMO/PP/CFG_GPOS-BT_V1.0.pdf">https://www.niap-ccevs.org/MMO/PP/CFG_GPOS-BT_V1.0.pdf</a>
FIPS186-4	<b>Digital Signature Standard (DSS)</b> Date 2013-07-19 Location <a href="https://csrc.nist.gov/pubs/fips/186-4/final">https://csrc.nist.gov/pubs/fips/186-4/final</a>
FIPS198-1	<b>The Keyed-Hash Message Authentication Code (HMAC)</b> Date 2008-07-16 Location <a href="https://csrc.nist.gov/pubs/fips/198-1/final">https://csrc.nist.gov/pubs/fips/198-1/final</a>
OSPPv4.2.1	<b>Protection Profile for General Purpose Operating Systems Version 4.2.1</b> Version 4.2.1 Date 2019-04-22 Location <a href="https://www.niap-ccevs.org/MMO/PP/PP_OS_V4.2.1.pdf">https://www.niap-ccevs.org/MMO/PP/PP_OS_V4.2.1.pdf</a>
RFC8017	<b>PKCS #1: RSA Cryptography Specifications Version 2.2</b> Author(s) K. Moriarty, B. Kaliski, J. Jonsson, A. Rusch Date 2016-11-01 Location <a href="http://www.ietf.org/rfc/rfc8017.txt">http://www.ietf.org/rfc/rfc8017.txt</a>

SP800-38A	<b>Recommendation for Block Cipher Modes of Operation: Methods and Techniques</b> Date 2001-12-01 Location <a href="https://csrc.nist.gov/pubs/sp/800/38/a/final">https://csrc.nist.gov/pubs/sp/800/38/a/final</a>
SP800-38C	<b>Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality</b> Date 2007-07-20 Location <a href="https://csrc.nist.gov/pubs/sp/800/38/c/upd1/final">https://csrc.nist.gov/pubs/sp/800/38/c/upd1/final</a>
SP800-38D	<b>Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC</b> Date 2007-11-28 Location <a href="https://csrc.nist.gov/pubs/sp/800/38/d/final">https://csrc.nist.gov/pubs/sp/800/38/d/final</a>
SP800-56A-Rev3	<b>Recommendation for Pair-Wise Key-Establishment Schemes Using Discrete Logarithm Cryptography</b> Date 2018-04-16 Location <a href="https://csrc.nist.gov/pubs/sp/800/56/a/r3/final">https://csrc.nist.gov/pubs/sp/800/56/a/r3/final</a>
SP800-90A-Rev1	<b>Recommendation for Random Number Generation Using Deterministic Random Bit Generators</b> Date 2015-06-24 Location <a href="https://csrc.nist.gov/pubs/sp/800/90/a/r1/final">https://csrc.nist.gov/pubs/sp/800/90/a/r1/final</a>

# A Appendixes

## A.1 Hardware Platforms Covered by this Evaluation

This evaluation covers the following hardware platforms.

For brevity, the processor manufacturer names were left out of the table leaving only the processor names. The Apple silicon processors start with the letter M. The Intel® processors start with either Core™ or Xeon®. Bluetooth is abbreviated as BT.

**Table 16: Hardware platforms**

Marketing Name	Model	Model Identifier	Processor	Micro Architecture	Security Chip	BT version	BT Chip
<b>2023</b>							
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
<b>2022</b>							
MacBook Pro (13-inch, M2, 2022)	A2338	Mac14,7	M2	ARMv8.6-A	SEP v2.0	5.0	4378
MacBook Air (M2, 2022)	A2861	Mac14,2	M2	ARMv8.6-A	SEP v2.0	5.0	4387
Mac Studio	A2615	Mac13,2	M1 Ultra	ARMv8.5-A	SEP v2.0	5.0	4387
	A2615	Mac13,1	M1 Max	ARMv8.5-A	SEP v2.0	5.0	4387
<b>2021</b>							
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
<b>2020</b>							
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	4364
MacBook Air (Retina, 13-inch, 2020)	A2179	MacBookAir9,1	Core i5-1030NG7 Core i7-1060NG7	Ice Lake	T2	5.0	4377

Marketing Name	Model	Model Identifier	Processor	Micro Architecture	Security Chip	BT version	BT Chip
MacBook Pro (13-inch, 2020, Four Thunderbolt 3 ports)	A2251	MacBookPro16,2	Core i5-1038NG7 Core i7-1068NG7	Ice Lake	T2	5.0	4377
MacBook Pro (13-inch, 2020, Two Thunderbolt 3 ports)	A2289	MacBookPro16,3	Core i5-8257U Core i7-8557U	Coffee Lake	T2	5.0	4377
iMac (Retina 5K, 27-inch, 2020)	A2115	iMac20,1 iMac20,2	Core i5-10500 Core i5-10600 Core i7-10700K Core i9-10910	Comet Lake	T2	5.0	4364
<b>2019</b>							
MacBook Air (Retina, 13-inch, 2019)	A1932	MacBookAir8,2	Core i5-8210Y	Amber Lake	T2	4.2	4355
MacBook Pro (13-inch, 2019, Four Thunderbolt 3 ports)	A1989	MacBookPro15,2	Core i5-8279U Core i7-8569U	Coffee Lake	T2	5.0	4364
MacBook Pro (13-inch, 2019, Two Thunderbolt 3 ports)	A2159	MacBookPro15,4	Core i5-8257U Core i7-8557U	Coffee Lake	T2	5.0	4377
MacBook Pro (15-inch, 2019)	A1990	MacBookPro15,1 MacBookPro15,3	Core i7-9750H Core i9-9880H	Coffee Lake	T2	5.0	4364
MacBook Pro (16-inch, 2019)	A2141	MacBookPro16,1 MacBookPro16,4	Core i7-9750H Core i9-9880H	Coffee Lake	T2	5.0	4377
Mac Pro (2019)	A1991	MacPro7,1	Xeon W-3223 Xeon W-3235 Xeon W-3245 Xeon W-3265M Xeon W-3275M	Cascade Lake	T2	5.0	4364
Mac Pro (2019 Rack)	A2304	MacPro7,1	Xeon W-3223 Xeon W-3235 Xeon W-3245 Xeon W-3265M Xeon W-3275M	Cascade Lake	T2	5.0	4364
<b>2018</b>							
MacBook Air (Retina, 13-inch, 2018)	A1932	MacBookAir8,1	Core i5-8210Y	Amber Lake	T2	4.2	4355
Mac mini (2018)	A1993	Macmini8,1	Core i5-8500B Core i7-8700B	Coffee Lake	T2	5.0	4364
MacBook Pro (15-inch, 2018)	A1990	MacBookPro15,1 MacBookPro15,3	Core i7-8750H Core i7-8850H Core i9-8950HK	Coffee Lake	T2	5.0	4364
MacBook Pro (13-inch, 2018, Four Thunderbolt 3 ports)	A1989	MacBookPro15,2	Core i5-8259U Core i7-8559U	Coffee Lake	T2	5.0	4364

Marketing Name	Model	Model Identifier	Processor	Micro Architecture	Security Chip	BT version	BT Chip
<b>2017</b>							
iMac Pro (2017)	A1862	iMacPro1,1	Xeon W-2140B Xeon W-2150B Xeon W-2170B Xeon W-2190B	Skylake	T2	5.0	4364

## A.2 SFR to CAVP Mapping Table

The CAVP certificates contain several different SoCs and micro-architectures in the operational environment (OE). The relationship between the SoCs and micro-architectures used by the devices claimed in this evaluation are specified in [Appendix A.1](#).

The following convention is used in the tables of this appendix to identify the cryptographic modules.

### KRN

- Apple corecrypto Module v13.0 [Apple ARM, Kernel, Software, SL1], or
- Apple corecrypto Module v13.0 [Intel, Kernel, Software, SL1]

### SEP

- SEP Hardware v2.0 in Apple silicon, or
- SEP Hardware v2.0 in Apple T2

### SKS

- Apple corecrypto Module v13.0 [Apple silicon, Secure Key Store, Hardware, SL2], or
- Apple corecrypto Module v13.0 [Apple ARM, Secure Key Store, Hardware, SL2]

### USR

- Apple corecrypto Module v13.0 [Apple ARM, User, Software, SL1], or
- Apple corecrypto Module v13.0 [Intel, User, Software, SL1]




### BC

The Broadcom core hardware module implementation names are:

- Crypto Hardware Module aes\_core\_gcm.vhd for Broadcom chip models 4378, 4387 and 4388.
- Cryptographic Hardware Module for Broadcom chip models 4355, 4364 and 4377.

The **T2** is marketed as Apple ARM technology and runs T2OS 13.

**Table 17: Cryptographic algorithm table**

SFR	Algorithm	Capabilities	Mod	Implementation	CAVP
FCS_CKM.1	RSA KeyGen [FIPS186-4] 	2048, 3072, 4096 (Apple silicon)	USR	vng_ltc	<a href="#">A3488</a>
		2048, 3072, 4096 (Intel)	USR	c_avx2	<a href="#">A3506</a>
	ECDSA KeyGen [FIPS186-4] 	P-256, P-384, P-521 (Apple silicon)	USR	vng_ltc	<a href="#">A3488</a>
		P-256, P-384, P-521 (Intel)	USR	c_avx2	<a href="#">A3506</a>
FCS_CKM.2	RSA Key Establishment [RFC8017] 	2048, 3072, 4096 (Apple silicon)	USR	c_ltc	Tested by the CCTL following the testing
		2048, 3072, 4096 (Intel)	USR	c_ltc	

SFR	Algorithm	Capabilities	Mod	Implementation	CAVP				
					Assurance Activity for FCS_CKM.2				
	ECC Key Establishment (KAS-ECC-SSC Sp800-56Ar3) <a href="#">[SP800-56A-Rev3]</a>	P-256, P-384, P-521 (Apple silicon)	USR	c_ltc	<a href="#">A3486</a>				
		P-256, P-384, P-521 (Intel)	USR	c_ltc	<a href="#">A3509</a>				
FCS_COP.1(1)	AES-CBC <a href="#">[SP800-38A]</a>	128-bit, 256-bit (Apple silicon)	USR	asm_arm	<a href="#">A3483</a>				
		128-bit, 256-bit (Intel)	USR	asm_aesni	<a href="#">A3501</a>				
	AES-GCM <a href="#">[SP800-38D]</a>	128-bit, 256-bit (Apple silicon)	USR	vng_asm	<a href="#">A3487</a>				
		128-bit, 256-bit (Intel)	USR	vng_asm	<a href="#">A3510</a>				
	AES-CCM <a href="#">[SP800-38C]</a>	128-bit		BC	4388	<a href="#">AES 5926</a>			
					4387	<a href="#">AES 5926</a>			
					4378	<a href="#">AES 5926</a>			
					4377	<a href="#">AES 4791</a>			
					4364	<a href="#">AES 3678</a>			
					4355	<a href="#">AES 3678</a>			
FCS_COP.1(2)	SHS Byte-oriented mode <a href="#">[FIPS186-4]</a>	SHA-1, SHA-256, SHA-384, SHA-512 (Apple silicon)	USR	vng_ltc	<a href="#">A3488</a>				
					SHA-256 (Apple silicon)	KRN	vng_ltc	<a href="#">A3521</a>	
						SKS	vng_ltc	<a href="#">A4259</a>	
					SHA-1, SHA-256, SHA-384, SHA-512 (Intel)	USR	c_avx2	<a href="#">A3506</a>	
						SHA-256 (Intel)	KRN	c_avx2	<a href="#">A3623</a>
							SKS	vng_neon	<a href="#">A4110</a>

SFR	Algorithm	Capabilities	Mod	Implementation	CAVP
		(T2)			
FCS_COP.1(3)	RSA SigVer <a href="#">[FIPS186-4]</a>	Modulo: 2048, 3072, 4096 with: SHA-1, SHA-256, SHA-384, SHA-512  (Apple silicon)	USR	vng_ltc	<a href="#">A3488</a>
		Modulo: 2048, 3072, 4096 with: SHA-1, SHA-256, SHA-384, SHA-512  (Intel)	USR	c_avx2	<a href="#">A3506</a>
		Modulo: 4096 with: SHA-256  (Intel)	KRN	c_avx2	<a href="#">A3623</a>
		Modulo: 4096 with: SHA-256  (T2)	SKS	vng_ltc	<a href="#">A4109</a>
	ECDSA SigVer <a href="#">[FIPS186-4]</a>	P-256, P-384, P-521 with: SHA-1, SHA-256, SHA-384, SHA-512  (Apple silicon)	USR	vng_ltc	<a href="#">A3488</a>
		P-521 with: SHA-512  (Apple silicon)	KRN	vng_ltc	<a href="#">A3521</a>
			SKS	vng_ltc	<a href="#">A4259</a>
		P-256, P-384, P-521 with: SHA-1, SHA-256, SHA-384, SHA-512  (Intel)	USR	vng_ltc	<a href="#">A3506</a>
FCS_COP.1(4)	HMAC Byte-oriented mode <a href="#">[FIPS198-1]</a>	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512  (Apple silicon)	USR	vng_ltc	<a href="#">A3488</a>
		HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512  (Intel)	USR	c_avx2	<a href="#">A3506</a>
FCS_RBG_EXT.1	CTR_DRBG	AES-256	USR	vng_asm	<a href="#">A3487</a>
			SEP	M2 Max (skg)	<a href="#">A3490</a>



SFR	Algorithm	Capabilities	Mod	Implementation	CAVP
	[SP800-90A-Rev1] <a href="#">pdf</a>	(Apple silicon)		M2 Pro (skg) M2 (skg) M1 Ultra (skg) M1 Max (skg) M1 Pro (skg)	
				M1 (skg)	<a href="#">A1362</a>
		AES-256 (Intel)	USR	vng_asm	<a href="#">A3510</a>
		AES-256 (T2)	SEP	T2 (skg)	<a href="#">DRBG 2029</a>

The following table shows the full coverage of CAVP tests for the Apple silicon models used in the devices covered by this evaluation and specified in Appendix A.1 .

**Table 18: Coverage of CAVP certificates for Apple silicon**

SoC	Micro Architecture	Cryptographic Modules			
		USR	KRN	SKS	SEP
M1	ARMv8.5-A	<a href="#">A3483</a> , <a href="#">A3486</a> , <a href="#">A3487</a> , <a href="#">A3488</a>	<a href="#">A3521</a>	<a href="#">A4259</a>	<a href="#">A1362</a>
M1 Max	ARMv8.5-A	<a href="#">A3483</a> , <a href="#">A3486</a> , <a href="#">A3487</a> , <a href="#">A3488</a>	<a href="#">A3521</a>	<a href="#">A4259</a>	<a href="#">A3490</a>
M1 Pro	ARMv8.5-A	<a href="#">A3483</a> , <a href="#">A3486</a> , <a href="#">A3487</a> , <a href="#">A3488</a>	<a href="#">A3521</a>	<a href="#">A4259</a>	<a href="#">A3490</a>
M1 Ultra	ARMv8.5-A	<a href="#">A3483</a> , <a href="#">A3486</a> , <a href="#">A3487</a> , <a href="#">A3488</a>	<a href="#">A3521</a>	<a href="#">A4259</a>	<a href="#">A3490</a>
M2	ARMv8.6-A	<a href="#">A3483</a> , <a href="#">A3486</a> , <a href="#">A3487</a> , <a href="#">A3488</a>	<a href="#">A3521</a>	<a href="#">A4259</a>	<a href="#">A3490</a>
M2 Max	ARMv8.6-A	<a href="#">A3483</a> , <a href="#">A3486</a> , <a href="#">A3487</a> , <a href="#">A3488</a>	<a href="#">A3521</a>	<a href="#">A4259</a>	<a href="#">A3490</a>
M2 Pro	ARMv8.6-A	<a href="#">A3483</a> , <a href="#">A3486</a> , <a href="#">A3487</a> , <a href="#">A3488</a>	<a href="#">A3521</a>	<a href="#">A4259</a>	<a href="#">A3490</a>

The following table shows the coverage of CAVP tests for the Intel processors used in the devices covered by this evaluation and specified in Appendix A.1. For those processor models not tested, the last column indicates the equivalent processor on which the CAVP tests were performed. The equivalence argument for these processors is that the reference testing is performed on a processor of the same Intel Micro Architecture and Intel processor Generation.

**Table 19: Coverage of CAVP certificates for Intel Processors**

Intel Processor	Gen	Micro Architecture	Cryptographic Modules		Equivalent processor
			USR	KRN	
Xeon W-2140B	W	Skylake	<a href="#">A3501</a> , <a href="#">A3506</a> ,	<a href="#">A3623</a>	Tested

Intel Processor	Gen	Micro Architecture	Cryptographic Modules		Equivalent processor
			USR	KRN	
			<a href="#">A3509</a> , <a href="#">A3510</a>		
Xeon W-2150B	W	Skylake			Xeon W-2140B
Xeon W-2170B	W	Skylake			Xeon W-2140B
Xeon W-2190B	W	Skylake			Xeon W-2140B
Xeon W-3223	W	Cascade Lake	<a href="#">A3501</a> , <a href="#">A3506</a> , <a href="#">A3509</a> , <a href="#">A3510</a>	<a href="#">A3623</a>	Tested
Xeon W-3235	W	Cascade Lake			Xeon W-3223
Xeon W-3245	W	Cascade Lake			Xeon W-3223
Xeon W-3265	W	Cascade Lake			Xeon W-3223
Xeon W-3265M	W	Cascade Lake			Xeon W-3223
Xeon W-3275M	W	Cascade Lake			Xeon W-3223
Core i5-8210Y	8 <sup>th</sup>	Amber Lake	<a href="#">A3501</a> , <a href="#">A3506</a> , <a href="#">A3509</a> , <a href="#">A3510</a>	<a href="#">A3623</a>	Tested
Core i5-8257U	8 <sup>th</sup>	Coffee Lake	<a href="#">A3501</a> , <a href="#">A3506</a> , <a href="#">A3509</a> , <a href="#">A3510</a>	<a href="#">A3623</a>	Tested
Core i5-8259U	8 <sup>th</sup>	Coffee Lake			Core i5-8257U
Core i5-8279U	8 <sup>th</sup>	Coffee Lake			Core i5-8257U
Core i7-8557U	8 <sup>th</sup>	Coffee Lake			Core i5-8257U
Core i7-8559U	8 <sup>th</sup>	Coffee Lake			Core i5-8257U
Core i7-8569U	8 <sup>th</sup>	Coffee Lake			Core i5-8257U
Core i5-8500B	8 <sup>th</sup>	Coffee Lake			Core i7-8700B
Core i7-8700B	8 <sup>th</sup>	Coffee Lake	<a href="#">A3501</a> , <a href="#">A3506</a> , <a href="#">A3509</a> , <a href="#">A3510</a>	<a href="#">A3623</a>	Tested
Core i7-8750H	8 <sup>th</sup>	Coffee Lake			Core i7-8700B
Core i7-8850H	8 <sup>th</sup>	Coffee Lake			Core i7-8700B
Core i9-8950HK	8 <sup>th</sup>	Coffee Lake			Core i7-8700B
Core i7-9750H	9 <sup>th</sup>	Coffee Lake			Core i9-9880H
Core i9-9880H	9 <sup>th</sup>	Coffee Lake	<a href="#">A3501</a> , <a href="#">A3506</a> , <a href="#">A3509</a> , <a href="#">A3510</a>	<a href="#">A3623</a>	Tested
Core i9-9880HK	9 <sup>th</sup>	Coffee Lake			Core i9-9880H
Core i5-10500	10 <sup>th</sup>	Comet Lake			Core i7-10700K
Core i5-10600	10 <sup>th</sup>	Comet Lake			Core i7-10700K
Core i7-10700K	10 <sup>th</sup>	Comet Lake	<a href="#">A3501</a> , <a href="#">A3506</a> , <a href="#">A3509</a> , <a href="#">A3510</a>	<a href="#">A3623</a>	Tested
Core i9-10910	10 <sup>th</sup>	Comet Lake	<a href="#">A3501</a> , <a href="#">A3506</a> , <a href="#">A3509</a> , <a href="#">A3510</a>	<a href="#">A3623</a>	Tested

Intel Processor	Gen	Micro Architecture	Cryptographic Modules		Equivalent processor
			USR	KRN	
Core i5-1030NG7	10 <sup>th</sup>	Ice Lake			Core i7-1060NG7
Core i5-1038NG7	10 <sup>th</sup>	Ice Lake			Core i7-1060NG7
Core i7-1060NG7	10 <sup>th</sup>	Ice Lake	<a href="#">A3501</a> , <a href="#">A3506</a> , <a href="#">A3509</a> , <a href="#">A3510</a>	<a href="#">A3623</a>	Tested
Core i7-1068NG7	10 <sup>th</sup>	Ice Lake			Core i7-1060NG7

The following table shows the full coverage of CAVP tests for the Apple T2 Security Chip, used as the security chip in devices using Intel processors, as specified in [Appendix A.1](#).

**Table 20: Coverage of CAVP certificates for Apple T2 Security Chip**

SoC	Micro Architecture	Cryptographic Modules	
		SKS	SEP
T2	ARMv8.1-A	<a href="#">A4109</a> , <a href="#">A4110</a>	<a href="#">DRBG 2029</a>

The following table shows the full coverage of CAVP tests for the different models of the Broadcom Chip with Bluetooth, used for bluetooth functionality, as specified in [Appendix A.1](#).

**Table 21: Coverage of CAVP certificates for Broadcom Chip with Bluetooth**

Broadcom Chip Model	Bluetooth version	Cryptographic Modules
		BC
4355	4.2	<a href="#">AES 3678</a>
4364	5.0	<a href="#">AES 3678</a>
4377	5.0	<a href="#">AES 4791</a>
4378	5.0	<a href="#">AES 5926</a>
4387	5.0	<a href="#">AES 5926</a>
4388	5.3	<a href="#">AES 5926</a>