Apple iPad and iPhone Mobile Devices with iOS 12

Common Criteria Configuration Guide

PP_MD_V3.1
with
EP_MDM_AGENT_V3.0,
PP_WLAN_CLI_EP_V1.0,
MOD_VPN_CLI_V2.1

Version 1.7
2019-03-12
VID: 10937

Prepared for:
Apple Inc.
One Apple Park Way
MS 927-1CPS
Cupertino, CA 95014
www.apple.com

Prepared by:
atsec information security Corp.
9130 Jollyville Road, Suite 260
Austin, TX 78759
www.atsec.com

Table of Contents

Table of Figures

Table of Tables

Revision History

Version	Date	Change
1.7	2019-03-12	Final revision

The following terms are trademarks of Apple Inc. in the United States, other countries, or both.

- AirPrint®
- App Store®
- Apple®
- Apple Pay®
- Apple Store®
- Cocoa®
- Cocoa Touch®
- Face ID®
- iCloud®
- iPad®
- iPad Air®
- iPad mini™
- iPad Pro®
- iPhone®
- iTunes®
- Keychain®
- Lightning®
- macOS®
- OS X®
- Safari®
- Touch ID®
- Xcode®

The following term is a trademark of Cisco in the United States, other countries, or both.

- IOS®

Common Criteria is a registered trademark of the National Security Agency, a federal agency of the United States.

1 Introduction

This document is written for administrators and users of Apple mobile devices that are managed using a mobile device management (MDM) solution. The Apple iPad and iPhone Mobile Devices with iOS 12 Security Target [ST] includes specifications for security where the mobile device operating environment includes a Wi-Fi network and includes evaluation of the Always-On virtual private network (VPN) provided by iOS.

According to the [ST], the evaluated devices are a series of Apple iPad and iPhone mobile devices running the iOS 12 operating system. The operating system manages the mobile device hardware, provides mobile device agent functionality, and provides the technologies required to implement native applications (apps). iOS 12 provides a built-in MDM application programming interface (API), giving management features that may be utilized by external MDM solutions and allowing enterprises to use Configuration Profiles to control some of the mobile device settings. The devices provide a consistent set of capabilities allowing for supervision. These capabilities include the preparation of devices for deployment, the subsequent management of the devices, and the termination of management.

The devices are expected to be part of an MDM solution that enables the enterprise to control and administer all devices that are part of the enterprise MDM solution.

The devices do not include the user apps that run on top of the operating system but do include controls that limit application behavior.

For the user, iOS 12 provides support to end users by providing support for connectivity using the Wireless LAN radio client, and provides functionality for the management of the Wi-Fi interface. Additionally, iOS 12 supports end users in an enterprise setting by providing always-on connectivity via an IPsec VPN tunnel in order to provide secure, reliable access to enterprise assets.

For clarity the following conventions will be used throughout this document.

- Keys: This document will specify keys, or attributes, that will need to be set to certain values to configure the mobile devices into the evaluated configuration. When a key is mentioned, it will be written in the following font: *AlwaysOn*.
- GUI navigation: There are certain configurations or values that can be viewed by navigating to it on the mobile device itself. When instructions for these are mentioned, it will be written in the following font: *Settings » Siri & Search.*
- Document sections: In the referenced Apple documentation the navigation to relevant sections are indicated as “Mobile Device Management”.

1.1 Purpose

This document is intended to provide information for the secure installation and use of the Target of Evaluation (TOE) for the Common Criteria (CC) evaluation of the mobile devices. The TOE was the mobile devices specified in Table 2: Mobile Devices Covered by the Evaluation. Readers of this document may use the term “mobile device” synonymously with the term “TOE”. This guidance is based on the CC requirements and the requirements given in the following documents:

- Protection Profile for Mobile Device Fundamentals, Version 3.1, dated 16 June, 2017 [PP_MD_V3.1],

- Extended Package for Mobile Device Management Agents, Version 3.0, dated 21 November, 2016 [EP_MDM_AGENT],
- Extended Package (EP) for Wireless LAN (WLAN) Clients, Version 1.0, dated 11 February, 2016 [PP_WLAN_CLI_EP], and
- PP-Module for Virtual Private Network (VPN) Clients, Version 2.1, dated 5 October, 2017 [MOD_VPN_CLI].

1.2 TOE Security Functionality

In the evaluated configuration, the mobile devices provide the following security functionality.

- Security audit
- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TOE Security Functionality (TSF)
- TOE access
- Trusted path/channels

1.3 Supporting Apple Documentation

This document provides clarifications to the Apple documentation as related to configuring the mobile devices into the evaluated configuration. The official Apple documentation should be referred to and followed only as directed within this document. This document supplements and supersedes the Apple documentation. Table 1: Guidance Documents lists the guidance documents relevant to the configuration and operation of the mobile devices.

Reference	Document Name	Location
Mobile Device Administrator Guidance		
[CC_GUIDE]	Apple iPad and iPhone Mobile Devices with iOS 12 Common Criteria Configuration Guide **(This document)**	https://www.niap-ccevs.org/st/st_vid10937-agd.pdf
[IOS_CFG] (2018-09-17)	Configuration Profile Reference	https://developer.apple.com/enterprise/documentation/Configuration-Profile-Reference.pdf
Mobile Device User Guidance		
[iPhone_UG]	iPhone User Guide for iOS 12 (2018)	https://help.apple.com/iphone/12/

Reference	Document Name	Location
[iPad_UG]	iPad User Guide for iOS 12 (2018)	https://help.apple.com/ipad/12/
[PASSCODE_Help] (June 8, 2018)	Use a passcode with your iPhone, iPad or iPod touch	https://support.apple.com/en-us/HT204060
Mobile Device Management		
[AConfig]	Apple Configurator Help (online guidance)	https://help.apple.com/configurator/mac/
[DEP_Guide] (12-2017)	Apple Deployment Programs Device Enrollment Program Guide	https://www.apple.com/business/docs/DEP_Guide.pdf
[PM_Help] (2018)	Profile Manager Help	https://help.apple.com/profilemanager/mac/
[IOS_MDM] (2018-09-17)	Mobile Device Management Protocol Reference	https://developer.apple.com/enterprise/documentation/MDM-Protocol-Reference.pdf
Supporting Documents		
[iOSDeployRef]	iOS Deployment Reference	https://help.apple.com/deployment/ios/
[IOS_LOGS]	Profiles and Logs	https://developer.apple.com/bug-reporting/profiles-and-logs/?platforms=ios
[LOGGING]	Logging	https://developer.apple.com/documentation/os/logging?language=objc
[MDM_SETTINGS_IT]	Mobile device management settings for IT	https://help.apple.com/deployment/mdm/
[TRUST_STORE]	List of available trusted root certificates in iOS 12, macOS 10.14, watchOS 5, and tvOS 12	https://support.apple.com/en-us/HT209144
[MANAGE_CARDS]	Manage the cards that you use with Apple Pay	https://support.apple.com/en-us/HT205583
[PAY_SETUP]	Set up Apple Pay	https://support.apple.com/en-us/HT204506
App Developer Guidance		

Reference	Document Name	Location
[CKTSREF] (2018)	Certificate, Key, and Trust Services	https://developer.apple.com/documentation/security/certificate_key_and_trust_services
[KEYCHAINPG] (2018)	Keychain Services Programming Guide	https://developer.apple.com/documentation/security/keychain_services
[IOS_SEC] (September 2018)	iOS Security (iOS 12)	https://www.apple.com/business/site/docs/iOS_Security_Guide.pdf

Table 1: Guidance Documents

1.4 Evaluated Mobile Devices

Table 2: Mobile Devices Covered by the Evaluation, lists the iPhone and iPad devices that are covered by the CC evaluation.

Processor	Device Name	Model Number
A8	iPhone 6	A1549
		A1586
		A1589
	iPhone 6 Plus	A1522
		A1524
		A1593
	iPad mini 4	A1538
		A1550
A8X	iPad Air 2	A1566
		A1567
A9	iPhone 6s	A1633
		A1688
		A1691 (China)
		A1700 (China)
	iPhone 6s Plus	A1634
		A1687
		A1690 (China)
		A1699 (China)
	iPhone SE	A1662
		A1723 (China)
		A1724 (China)

Processor	Device Name	Model Number
	iPad 9.7-inch (5th generation)	A1822
		A1823
A9X	iPad Pro 12.9-inch	A1584
		A1652
	iPad Pro 9.7-inch	A1673
		A1674
		A1675
A10 Fusion	iPhone 7	A1660
		A1779 (Japan)
		A1780 (China)
		A1778
	iPhone 7 Plus	A1661
		A1785 (Japan)
		A1786 (China)
		A1784
	iPad 9.7-inch (6th generation)	A1893
		A1954
A10X Fusion	iPad Pro 12.9-inch (2nd generation)	A1670
		A1671
		A1821 (China)
	iPad Pro 10.5-inch	A1701
		A1852 (China)
		A1709
A11 Bionic	iPhone 8	A1863
		A1906 (Japan)
		A1907
		A1905 (GSM)
	iPhone 8 Plus	A1864
		A1898 (Japan)
		A1899
		A1897 (GSM)
	iPhone X	A1865 (Japan)
		A1902 (Japan)
		A1903 (Japan)

Processor	Device Name	Model Number
		A1901
A12 Bionic	iPhone XS	A1920 (US/CA/HK)
		A2097
		A2098 (Japan)
		A2099 (Global)
		A2100 (China)
	iPhone XS Max	A1921 (US/CA)
		A2101 (Global)
		A2102 (Japan)
		A2103 (Global)
		A2104 (China/HK)
	iPhone XR	A1984 (US/CA)
		A2105 (Global)
		A2106 (Japan)
		A2107 (US/CA)
		A2108 (HK/China)
A12X Bionic	iPad Pro 11-inch	A1934 (US/CA)
		A1979 (China)
		A1980
		A2013 (US/CA)
	iPad Pro 12.9 inch	A2014 (US/CA)
		A1876
		A1895
		A1986 (China)

Table 2: Mobile Devices Covered by the Evaluation

1.5 Assumptions

The following assumptions apply when operating the mobile devices in the evaluated configuration. These assumptions must be valid within the organization to maintain security of the mobile devices.

1.5.1 Administrators

- One or more competent, trusted personnel who are not careless, willfully negligent, or hostile, are assigned and authorized as the mobile device administrators, and do so using and abiding by guidance documentation.
- Device administrators are trusted to follow and apply all administrator guidance in a trusted manner.

- Personnel configuring the mobile device and its operational environment will follow the applicable security configuration guidance.
- Device administrators will configure the mobile device's security functions correctly to create the intended security policy.

1.5.2 Mobile device users

- Mobile device users are not willfully negligent or hostile and use the mobile device within compliance of a reasonable enterprise security policy.
- The mobile device user exercises precautions to reduce the risk of loss or theft of the mobile device.
- The mobile device user will immediately notify the administrator if the mobile device is lost or stolen.
- Physical security, commensurate with the value of the mobile device and the data it contains, is assumed to be provided by the environment.

1.5.3 Network

- The mobile device relies on network connectivity to carry out its management activities. The mobile device will robustly handle instances when connectivity is unavailable or unreliable.
- Information cannot flow between the wireless client and the internal wired network software integrity verification of the MDM agent.
- Information cannot flow onto the network to which the VPN client's host is connected without passing through the device.

1.5.4 Other

- The MDM Agent relies upon mobile platform and hardware evaluated against the [PP_MD_V3.1] and assured to provide policy enforcement as well as cryptographic services and data protection. The mobile device platform provides trusted updates and software integrity verification of the MDM Agent.

1.6 Organizational Security Policies

The following requirements are for policies that must be implemented by the deploying organization in support of meeting the security requirements of the [ST].

- The mobile device administrators must adhere to the organizational security policies.
- The mobile device must be supervised using an MDM solution prior to connecting to the network.
- The mobile device user is held accountable for his/her actions while using the device.
- The mobile device user must promptly report his/her lost or stolen device to the mobile device administrator. The mobile device administrator must take appropriate actions using the MDM solution used to manage the mobile device.

1.7 Security Functional Requirements (SFRs) in the ST requiring configuration

In the evaluated configuration, the devices address each SFR in the following table. Table *3: SFR Configuration Requirements*, identifies each SFR specified in the Security Target [ST] and provides references to sections within this document for information on the function in the Related Section column. The Configurable? column denotes if the function needs to or can be configured.

SFR ID	Function Description	Configurable?	Related Section
FAU_ALT_EXT.2 {AGENT}	Agent Alerts	No	Section 2.3.5
FAU_GEN.1(1) {MDF}	Audit Data Generation	Yes	Section 4.1
FAU_GEN.1(2) {AGENT}	Audit Data Generation	Yes	Section 4.1
FAU_SEL.1(2) {AGENT}	Security Audit Event Selection	Yes	Section 4.2, Section 4.3
FAU_STG.1 {MDF}	Audit Storage Protection	No: Audit records are not accessible to device Administrators or Users and must be viewed on a trusted workstation or MDM server.	Section 4.2
FAU_STG.4 {MDF}	Prevention of Audit Data Loss	No: The default behavior is to overwrite the oldest entry.	Section 4.2
FCS_CKM.1(1) {MDF} {VPN} {AGENT}	Cryptographic Key Generation	No: The API allows specification of the requested key sizes and key types.	Section 3.2.1
FCS_CKM.1(2) {WLAN}	WLAN Cryptographic Key Generation (Symmetric Keys for WPA2 Connections	No: Wireless LAN (WLAN) keys are generated for the cipher suite offered by the access point.	N/A
FCS_CKM.1/VPN {VPN}	VPN Cryptographic Key Generation (IKE)	No: IKEv2 is an available option and the API allows for the specification of the key size and key types.	Section 3.3.6

SFR ID	Function Description	Configurable?	Related Section
FCS_CKM.2(1) {MDF} {VPN} {AGENT}	Cryptographic Key Establishment	No: The API allows specification of the requested key sizes and key types.	Section 3.2.2
FCS_CKM.2(2) {MDF}	Cryptographic Key Establishment (While device is locked)	No: Key establishment is hard coded.	Section 3.2.2
FCS_CKM.2/WLAN {WLAN}	WLAN Cryptographic Key Distribution (GTK)	No: The WLAN protocol is implemented according to IEEE 802.11 2012.	N/A
FCS_CKM_EXT.1 {MDF}	Cryptographic Key Support (REK)	No: REK is permanently etched in silicon and is both unmodifiable as well as inaccessible by iOS and apps.	N/A
FCS_CKM_EXT.2 {MDF}	Cryptographic Key Random Generation (DEK)	No: Generation and maintenance of DEK is hard coded.	N/A
FCS_CKM_EXT.3 {MDF}	Cryptographic Key Generation (KEK)	No: Generation and maintenance of KEK is hard coded.	N/A
FCS_CKM_EXT.4 {MDF} {VPN} {WLAN} {AGENT}	Key Destruction	No: Zeroization of keys is hard coded.	N/A
FCS_CKM_EXT.5 {MDF}	TSF Wipe	Yes	Section 3.4.3
FCS_CKM_EXT.6 {MDF}	Salt Generation	No: Generation and maintenance of Salt is hard coded.	N/A
FCS_CKM_EXT.7 {MDF}	Cryptographic Key Support (REK)	No: REK is permanently etched in silicon and is both unmodifiable as well as inaccessible by iOS and apps.	N/A

SFR ID	Function Description	Configurable?	Related Section
FCS_COP.1(1) {MDF} {VPN} {AGENT} {WLAN}	Confidentiality Algorithms	No: For AES operations performed by the TSF. No: For AES operations performed by third party where the API allows specification of the AES cipher type	Section 3.2.6
FCS_COP.1(2) {MDF} {VPN} {AGENT} {WLAN}	Hashing Algorithms	No: For hash operations performed by the TSF for TLS. Yes: For hash operations performed for VPN No: For hash operations performed by third party where the API allows specification of the hash cipher type.	Section 3.2.3
FCS_COP.1(3) {MDF} {VPN} {AGENT} {WLAN}	Signature Algorithms	No: For signature operations performed by TSF. No: For signature operations performed by third party where the API allows specification of the hash cipher type.	Section 3.2.1
FCS_COP.1(4) {MDF} {VPN} {AGENT} {WLAN}	Keyed Hash Algorithms	No: For HMAC operations performed by TSF No: For HMAC operations performed by third party where the API allows specification of the hash cipher type.	Section 3.2.3
FCS_COP.1(5) {MDF} {AGENT} {WLAN}	Password-Based Key Derivation Functions	No: Generation and maintenance of PBKDF is hard coded.	N/A

SFR ID	Function Description	Configurable?	Related Section
FCS_HTTPS_EXT.1 {MDF} {AGENT}	HTTPS protocol	No: The used HTTPS cipher suite is defined by the HTTPS server where all cipher suites listed in the [ST] are always available.	Section 3.3.2
FCS_IPSEC_EXT.1 {VPN}	IPsec	Yes	Section 3.3.5, Section 3.3.3
FCS_IV_EXT.1 {MDF}	Initialization Vector Generation	No: Generation and maintenance of IVs is hard coded.	N/A
FCS_RBG_EXT.1 {MDF} {VPN} {WLAN} {AGENT} (Kernel and User space and SEP iterations.)	Cryptographic Operation (Random Bit Generation)	No: Generation of random numbers is hard coded.	Section 3.2.4
FCS_SRV_EXT.1 {MDF}	Cryptographic Algorithm Services	No	Section 3.2
FCS_STG_EXT.1 {MDF}	Secure Key Storage	No	Section 3.6.12
FCS_STG_EXT.2 {MDF} {VPN}	Encrypted Cryptographic Key Storage DEK and KEK encryption	No: Generation and maintenance of DEK and KEK is hard coded.	N/A
FCS_STG_EXT.3 {MDF}	Integrity of Encrypted Key Storage	No: Generation and maintenance of DEK and KEK is hard coded.	N/A
FCS_STG_EXT.4 {AGENT}	Cryptographic Key Storage	No	N/A
FCS_TLSC_EXT.1 {MDF} {AGENT}	TLS Protocol	Yes	Section 3.3.2
FCS_TLSC_EXT.1/WLAN {WLAN}	Extensible Authentication Protocol-Transport Layer Security (EAP-TLS)	No: Used TLS cipher suites are defined by the TLS server where all cipher suites listed in the [ST] are always available. The API of the third-party application defines specific TLS protocol rules.	Section 3.3.1
FCS_TLSC_EXT.2 {MDF}	TLS Protocol	No	Section 3.3.2

SFR ID	Function Description	Configurable?	Related Section
FDP_ACF_EXT.1 {MDF}	Security Access Control	No: Access control settings are hard coded.	Section 3.4.2
FDP_DAR_EXT.1 {MDF}	Protected Data Encryption	No: Data is always encrypted. TSF is hard coded to use the appropriate data protection levels based on classes.	Section 3.4
FDP_DAR_EXT.2 {MDF}	Sensitive Data Encryption	No: Data is always encrypted. TSF is hard coded to use the appropriate data protection level based on classes.	Section 3.4
FDP_IFC_EXT.1 {MDF} {VPN}	Subset Information Flow Control	Yes	Section 3.3.5, Section 3.3.3
FDP_PBA_EXT.1 {MDF}	Storage of Critical Biometric Parameters	No	N/A
FDP_RIP.2 {VPN}	Full Residual Information Protection	No	N/A
FDP_STG_EXT.1 {MDF}	User Data Storage	No: The trust anchor database maintenance is hard coded. The mobile device administrator can add/remove their own Anchors of Trust to/from that database.	Section 3.5.6
FDP_UPC_EXT.1 {MDF}	Inter-TSF User Data Transfer Protection	Yes: Depending on the protocol used, configuration is possible (e.g., IPsec) while other options are not configurable (e.g., Bluetooth)	FTP_ITC_EXT.1 (Section 3.3) FCS_TLSC_EXT.1 (Section 3.3.2) FCS_IPSEC_EXT.1 (Section 3.3.5) FCS_HTTPS_EXT.1 (Section 3.3.2)
FIA_AFL_EXT.1 {MDF}	Authentication Failure Handling	Yes	Section 3.5.4

SFR ID	Function Description	Configurable?	Related Section
FIA_BLT_EXT.1 {MDF}	Bluetooth User Authorization	No: The Bluetooth protocol allows different types of authorization which are supported by the mobile device. The used authorization type depends on the remote device capability.	Section 3.3.4
FIA_BLT_EXT.2 {MDF}	Bluetooth Mutual Authentication	No: Bluetooth mutual authentication is required prior to data transfer.	Section 3.3.4
FIA_BLT_EXT.3 {MDF}	Rejection of Duplicate Bluetooth Connections	No: No mobile device can establish duplicative Bluetooth connections.	N/A
FIA_BLT_EXT.4 {MDF}	Secure Simple Pairing	No: Secure simple pairing cannot be disabled.	Section 3.3.4
FIA_BMG_EXT.1 {MDF}	Accuracy of Biometric Authentication	No	N/A
FIA_BMG_EXT.2 {MDF}	Biometric Enrollment	No	Section 3.5.3
FIA_BMG_EXT.3 {MDF}	Biometric Verification	No	Section 3.5.3
FIA_BMG_EXT.5 {MDF}	Handling Unusual Biometric Templates	No	N/A
FIA_ENR_EXT.2 {AGENT}	Enrollment of Mobile Device into Management	Yes	Section 2.3.1
FIA_PAE_EXT.1 {WLAN}	Port Access Entity (PAE) Authentication	No: The WLAN protocol is implemented according to IEEE 802.11 2012.	N/A
FIA_PMG_EXT.1 {MDF}	Password Management	Yes	Section 3.5.1
FIA_TRT_EXT.1 {MDF}	Authentication Throttling	No: The authentication delay is hard coded.	N/A

SFR ID	Function Description	Configurable?	Related Section
FIA_UAU.5 {MDF}	Multiple Authentication Mechanisms	Yes	Section 3.5.3
FIA_UAU.6(1) {MDF}	Re-Authentication	No: Users must be re-authenticated before any changes to the password authentication factor can be made.	Section 3.5.3
FIA_UAU.6(2) {MDF}	Re-Authentication (Locked)	No	Section 3.5.5
FIA_UAU.7 {MDF}	Protected Authentication Feedback	No: Enabled by default.	Section 3.5.2
FIA_UAU_EXT.1 {MDF}	Authentication for Cryptographic Operations	Yes: The mobile device user must set a passphrase to enable authentication token protection.	Section 3.5.1
FIA_UAU_EXT.2 {MDF}	Timing of Authentication	No	Section 3.6.2
FIA_X509_EXT.1 {MDF} {VPN} {AGENT}	Validation of Certificates	No: The certificate validation rules are hard coded.	N/A
FIA_X509_EXT.2 {MDF} {VPN} {AGENT}	X509 Certificate Authentication	Yes: The certificates required for authentication must be provided.	Section 3.3, 3.5.6
FIA_X509_EXT.2/ WLAN {WLAN}	X509 Certificate Authentication (EAP-TLS)	Yes	Section 3.5.6
FIA_X509_EXT.3 {MDF} {AGENT}	Request Validation of Certificates	No: The API is provided with certificate validation rules hard coded.	Section 3.5.6
FMT_MOF_EXT.1 {MDF}	Management of Security Functions Behavior	Yes	Section 1.8
FMT_POL_EXT.2 {AGENT}	Trusted Policy Update	No	N/A
FMT_SMF_EXT.1 {MDF}	Specification of Management Functions	Yes	Section 1.8

SFR ID	Function Description	Configurable?	Related Section
FMT_SMF_EXT.1 / WLAN {WLAN}	Specification of Management Functions (WLAN)	Yes	Section 1.8
FMT_SMF.1/VPN 1 {VPN}	Specification of Management Functions (VPN)	Yes	Section 1.8
FMT_SMF_EXT.2 {MDF}	Specification of Remediation Actions	Yes	Section 2.3.4, 3.4.3
FMT_SMF_EXT.3 {AGENT}	Specification of Management Functions (Agent)	No	N/A
FMT_UNR_EXT.1 {AGENT}	User Unenrollment Prevention	Yes	Section 2.3.4
FPT_AEX_EXT.1 {MDF}	Anti-Exploitation Services (ASLR)	No: The service is hard coded.	N/A
FPT_AEX_EXT.2 {MDF}	Anti-Exploitation Services (Memory Page Permissions)	No: The service is hard coded.	N/A
FPT_AEX_EXT.3 {MDF}	Anti-Exploitation Services (Overflow Protection)	No: The service is hard coded.	N/A
FPT_AEX_EXT.4 {MDF}	Domain Isolation	No: The service is hard coded.	N/A
FPT_JTA_EXT.1 {MDF}	JTAG Disablement	No: JTAG interfaces are not present on iOS devices.	N/A
FPT_KST_EXT.1 {MDF}	Key Storage	No: Keys are stored in secure enclave or in key chain. Wrapped keys are stored in Effaceable Storage.	N/A
FPT_KST_EXT.2 {MDF}	No Key Transmission	No: Keys are stored in secure enclave or in key chain.	N/A
FPT_KST_EXT.3 {MDF}	No Plaintext Key Export	No: Keys are stored in secure enclave that does not provide key export facility. The mobile device does not export keys stored in key chain.	N/A
FPT_NOT_EXT.1 {MDF}	Self-Test Notification	No	N/A

SFR ID	Function Description	Configurable?	Related Section
FPT_STM.1 {MDF}	Reliable Time Stamps	Yes	Section 3.6.4
FPT_TST_EXT.1 {MDF} {AGENT}	TSF Cryptographic Functionality Testing	No	Section 3.2
FPT_TST_EXT.1/ WLAN {WLAN}	TSF Cryptographic Functionality Testing (WLAN)	No	Section 3.2
FPT_TST_EXT.1/ VPN {VPN}	TSF Self-Test (VPN)	No	Section 3.2
FPT_TST_EXT.2 {MDF}	TSF Integrity Testing	No	N/A
FPT_TST_EXT.3 {MDF}	TSF Integrity Testing	No	Section 3.5.6
FPT_TUD_EXT.1 {MDF} {VPN}	Trusted Update: TSF Version Query	No	N/A
FPT_TUD_EXT.2 {MDF}	Trusted Update Verification	No	N/A
FPT_TUD_EXT.3 {MDF}	Trusted Update Verification	No	Section 3.6.9
FPT_TUD_EXT.4 {MDF}	Trusted Update Verification	No	Section 3.6.9
FTA_SSL_EXT.1 {MDF}	TSF and User-initiated Locked State	Yes	Section 3.6.3
FTA_TAB.1 {MDF}	Default TOE Access Banners	Yes	Section 3.6.5
FTA_WSE_EXT.1 {WLAN}	Wireless Network Access	Yes	Section 3.6.7
FTP_ITC_EXT.1(1) {VPN}	Trusted Channel Communication	Yes	Section 3.3
FTP_ITC_EXT.1(2) {AGENT} {MDF}	Trusted Channel Communication	Yes	Section 3.3
FTP_ITC_EXT.1/ WLAN (3) {WLAN}	Trusted Channel Communication	Yes	Section 3.3

Table 3: SFR Configuration Requirements

1.8 Security Management Configuration

In the evaluated configuration, the mobile devices perform the management functions listed in Table 4: Required Mobile Device Management Functions.

These management functions can be managed either by the mobile device user or by an authorized mobile device administrator (marked by ‘X’)

In addition, the Provided Guidance column references the section(s) in this document where guidance can be found to perform the respective management function. The management function values in parenthesis (e.g., F1, F2) in the following table correspond to the function values specified in the [ST] Table 4 and include the additional management functions specific to Wi-Fi and VPN management functionality also given in the [ST].

Management Function	Restricted to the User	Administrator	Restricted to the Administrator	Provided Guidance
Configure password policy (F1)	-	X	X	Section 3.5.1
Configure session locking policy (F2)	-	X	X	Sections 3.6.3
Enable/disable the VPN protection (F3)	-	X	X	Sections 3.5.3
Enable/disable Bluetooth, Wi-Fi, cellular radio, NFC (F4)	X	-	-	Section 3.6.7
Enable/disable cameras (F5)	X	X	-	Section 3.6.6
Enable/disable microphones (F5)	X	X	-	Section 3.6.6
Transition to the locked state (F6)	-	X	-	Section 3.6.3
TSF wipe of protected data (F7)	-	X	-	Section 3.4.3
Configure application installation policy by denying installation of applications (F8)	-	X	X	Section 3.6.11
Import keys/secrets into the secure key storage (F9)	-	X	-	Section 3.2.5

Management Function	Restricted to the User	Administrator	Restricted to the Administrator	Provided Guidance
Destroy imported keys/secrets and no other keys/secrets in the secure key storage (F10)	-	X	-	Section 3.2.5
Import X.509v3 certificates in the Trust Anchor Database (F11)	-	X	X	Section 3.5.6
Remove imported X509v3 certificates and no other X509v3 certificates in the Trust Anchor Database (F12)	X	X	-	Section 3.5.6
Enroll the mobile device in management (F13)	X	-	-	Section 2.3.1
Remove applications (F14)	-	X	X	Section 3.6.1
Update system software (F15)	-	X	X	Section 3.6
Install applications (F16)	-	X	X	Section 3.6.9
Remove Enterprise applications (F17)	-	X	-	Section 3.6.1
Configure the Bluetooth trusted channel[1] (F18)	X	-	-	Section 3.6.7
Enable/disable display notifications in the locked state of all notifications (F19)	X	X	-	Section 3.6.2

[1] There is no configuration for the Bluetooth trusted channel. It is secure by default.

Management Function	Restricted to the User	Administrator	Restricted to the Administrator	Provided Guidance
Enable data-at-rest protection (F20)	-	X	X	Section 3.4.1
Enable/disable location services (across device and on a per-app basis) (F22)	X	X	-	Section 3.6.8
Enable/disable the use of Biometric Authentication Factor (F23)	-	X	X	Section 3.5.3
Wipe Enterprise data (F28)	-	X	-	Section 3.4.3
Configure whether to establish a trusted channel or disallow establishment if the TSF cannot establish a connection to determine the validity of a certificate (F30)	-	X	-	Section 3.5.6
Configure certificate used to validate digital signature on application (F33)	-	X	X	Sections 3.5.6
Configure the unlock banner (F36)	-	X	X	Section 3.6.5
Configure the auditable items (F37)	-	X	X	Section 4.3
Enable/disable the Always On VPN protection (F45)	-	X	X	Section 3.3.5

Management Function	Restricted to the User	Administrator	Restricted to the Administrator	Provided Guidance
Configure security policy for Wi-Fi network (in FMT_SMF_EXT.1.1/WLAN {WLAN})	-	X	X	Section 3.3.6
Specify CA to accept certificates from (in FMT_SMF_EXT.1.1/WLAN {WLAN})	-	X	X	Section 3.3.6
Specify Wi-Fi security type (in FMT_SMF_EXT.1.1/WLAN {WLAN})	-	X	X	Section 3.3.6
Specify authentication protocol for Wi-Fi (in FMT_SMF_EXT.1.1/WLAN {WLAN})	-	X	X	Section 3.3.6
Specify client credentials used for Wi-Fi authentication (in FMT_SMF_EXT.1.1/WLAN {WLAN})	-	X	X	Section 3.3.6
Specify IPsec-capable network devices to use for connection) (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6
Specify client credentials for VPN connection (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6
Configure the reference identifier of the peer (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6
Configure IKE protocol versions used (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6
Configure IKE authentication techniques used (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6

Management Function	Restricted to the User	Administrator	Restricted to the Administrator	Provided Guidance
Configure cryptoperiod for established session keys (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6
Configure certificate revocation check (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6
Configure algorithm suites used during IPsec exchanges (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6
Load X.509v3 certificate for VPN security functions (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6
Update TOE and verify updates (in FMT_SMF.1/VPN {VPN})	-	X	X	Section 3.3.6

Table 4: Required Mobile Device Management Functions

1.9 Un-evaluated Functionalities

The following security functionalities were not evaluated and are therefore not included in the secure configuration of the mobile devices.

1.9.1 Two-Factor Authentication

Two-factor authentication is an extra layer of security for an Apple ID used in the Apple store, iCloud and other Apple services. It is designed to enhance the security on these on-line Apple accounts.

This feature is outside the scope of the evaluation.

1.9.2 Bonjour

Bonjour is Apple's standards-based, zero configuration network protocol that lets devices find services on a network.

This feature is outside the scope of the evaluation.

1.9.3 VPN Split Tunnel

VPN split tunnel is not included in the evaluation, and must be disabled in the mobile device configurations meeting the requirements of this CC evaluation.

1.9.4 Siri Interface

The Siri interface supports some commands related to configuration settings.

This feature is not included in the evaluation and must be disabled in the mobile device configurations that meet the requirements of this CC evaluation.

1.9.5 Shared iPad for education

Apple offers the ability to configure the iPad devices for multiple users. This configuration was not included in the evaluation and must not be used in the mobile device configurations that meet the requirements of this CC evaluation.

1.9.6 Third-party MDM Agents

Some third-party applications are available that provide functionality as a mobile device MDM Agent. No third-party MDM Agent applications were included in the evaluation and are outside the scope of the evaluated configuration.

1.9.7 VPN Protocols and Authentication Methods

The following Virtual Private Network (VPN) protocols are not included in the evaluation and must be disabled in the mobile device configurations that meet the requirements of this CC evaluation.

- Cisco IPsec
- Layer Two Tunneling Protocol (L2TP) over IPsec
- Secure Sockets Layer (SSL) VPN
- Shared secret authentication

2 Secure Delivery and Installation

2.1 Prerequisites

Prior to deploying the mobile device(s) onto the network, an MDM solution must be architected and deployed. The MDM solution will support the mobile device administrator in configuring and managing the mobile devices. There are various MDM solutions that can be used to achieve this.

A VPN gateway supporting IPsec and the necessary VPN settings discussed below must be architected and deployed. The VPN infrastructure will support secure communication with the devices. If the devices will be utilizing x509 certificates for authenticating to the VPN connection then a public key infrastructure (PKI) system will need to be deployed by the organization which includes a certificate authority (CA) trusted both by the VPN gateway and the device, and an Online Certificate Status Protocol (OCSP) responder or published certificate revocation list (CRL) to service revocation checking requests.

2.2 Secure Delivery of the Devices

The evaluated mobile devices are intended for authorized mobile device users of entities such as business organizations and government agencies.

The mobile device administrator of the devices is responsible for performing the necessary configuration to ensure that the mobile devices are configured as specified by the evaluation.

2.2.1 Obtaining the mobile device(s)

To obtain a device listed in Table 2: Mobile Devices Covered by the Evaluation, follow the directions for the distribution channel that best fits your situation.

The normal distribution channels for obtaining these devices include the following.

- The Apple Store (either a physical store or online at https://apple.com)
- Apple retailers
- Service carriers (e.g., AT&T, Verizon)
- Resellers

Business specific distribution channel

There is a distinct online store for Business customers with a link from the “Apple Store” to Apple and Business: (https://www.apple.com/business/). Additionally, the following link to “Shop for Business” is provided (https://www.apple.com/retail/business/).

Government specific distribution channel

Government customers can use the link: https://www.apple.com/r/store/government/.

Additional

Large customers can have their own Apple Store Catalog for their employees to purchase devices directly from Apple under their corporate employee purchase program.

2.2.2 Verifying the device(s)

When the mobile devices are received, the model of the devices should be verified to verify that the model number is one of those listed in Table 2: Mobile Devices Covered by the Evaluation. This can be accomplished using any of the following methods.

- Physically checking the back of the mobile devices.
- Once authenticated to the mobile device the information is available to mobile device users in *Settings » General » About*, and clicking on the number displayed next to “Model”.
- Mobile device administrators can query the mobile devices using the Mobile Device Management (MDM) protocol described in [IOS_MDM]. The Results Payload from the mobile device provides the requested information.

The iOS version of the devices, which must be a version of iOS 12, should also be verified. This can be accomplished using either of the following methods.

- A mobile device user can obtain information about the iOS software on the mobile device by following the instructions in the [iPad_UG] and the [iPhone_UG] in “Get information about your iPhone” or “Get information about your iPad”.
- Mobile device administrators can query the mobile devices using the MDM protocol described in [IOS_MDM]. The Results Payload from the mobile device provides the requested information.

2.3 Mobile Device Supervision and Configuration

In order to ensure that the devices are configured in a way that meets the requirements of this Common Criteria evaluation, the devices must be placed under management (supervised mode).

Once in supervised mode, the mobile devices are typically managed using an MDM solution. The process for doing this will vary based on the MDM solution chosen by the organization deploying the devices and it is up to the mobile device administrator to determine the detailed steps as they apply to the organization’s chosen MDM solution. The mobile devices are configured through the use of Configuration Profiles that are specified by the mobile device administrator and deployed to the mobile devices.

2.3.1 Mobile Device Enrollment into Management Configuration

iOS natively includes an MDM agent. Mobile device users and/or device administrators can enroll the mobile device in management. Information for enrolling the mobile device is provided in the section “Configuration and management, subsection Mobile device management (MDM)” of the [iOSDeployRef].

The MDM server identity is provided to the mobile device by sending an MDM payload in a Configuration Profile.

The methods by which the mobile device can be enrolled for management are as follows.

- The Device Enrollment Program (DEP), which provides an automated and enforced method of automatically enrolling new devices
- Using Apple’s Profile Manager, which provides a manual method of enrolling mobile devices

- Using the Apple Configurator 2, which provides both automated and manual methods of enrolling mobile devices
- Using Email or a Website, which provides a way to distribute an enrollment profile to a mobile device

2.3.1.1 Device Enrollment Program

For the DEP, each MDM server must be registered with Apple at the MDM server DEP management portal which is made available by Apple at https://deploy.apple.com.

The DEP provides details about the server entity to identify it uniquely throughout the organization deploying the MDM server. Each server can be identified by either its system-generated universally unique identifier (UUID) or by a user-provided name assigned by one of the organization's users. Both the UUID and server name must be unique within the organization.

The organization assigns iOS devices to Apple's virtual MDM server using either Apple order numbers or device serial numbers. When the iOS device is powered on, the mobile device will automatically connect to the virtual MDM server during setup and will be assigned to the MDM server specified in the MDM payload sent by the virtual MDM server to the iOS device.

During the mobile device enrollment, the MDM enrollment service returns a JavaScript Object Notation (JSON) dictionary to the mobile device with the keys shown in Table 5: Essential MDM Payload.

Additional information on the DEP is provided in the [DEP_Guide]. Additional information on managing mobile devices is provided in [IOS_MDM].

2.3.1.2 Apple Profile Manager

For enrolling a device using Apple's Profile Manager, see the "Mobile Device Management" section of [PM_Help].

2.3.1.3 Apple Configurator

For enrolling a device using the Apple Configurator 2, see the "Prepare devices using automated enrollment" and "Prepare devices manually" sections of [AConfig].

2.3.1.4 Other Methods

Other methods of enrollment may be specific to the MDM application being used by a deploying organization. In general, the Configuration Profile is made available to the mobile device often through a link provided on a website, or by email to the mobile device user. Once the mobile device user clicks the link the enrollment process is started.

2.3.2 Mobile Device Configuration

Many aspects of the security functionality of the mobile devices are configured using Configuration Profiles that are installed on the mobile devices. Configuration Profiles are Extensible Markup Language (XML) files that allow the distribution of configuration information to mobile devices. They may contain settings for a number of configurable parameters on the mobile device.

Configuration Profiles can be deployed in any one of the following ways.

- Using the Apple Configurator 2 tool, available from the Apple Store
- Via an email message
- Via a web page
- Using over-the-air configuration
- Using over the air configuration via a MDM application

iOS supports using encryption to protect the contents of Configuration Profiles, and Configuration Profiles can also be signed to guarantee data integrity.

Within a Configuration Profile, various Keys are used to specify the desired configuration. These are organized by topic into groups called “Payloads.”

Detailed information on Configuration Profiles is given in the Configuration Profile Reference [IOS_CFG].

The following mandatory configurations must be configured using Configuration Profiles.

2.3.3 Configure MDM Agent and MDM Communications

MDM Agent-Server communication is achieved securely using the MDM protocol which is built on top of HTTP, transport layer security (TLS), and push notifications that use HTTP PUT over TLS (secure sockets layer (SSL)). A managed mobile device uses an identity to authenticate itself to the MDM server over TLS (SSL). This identity can be included in the profile as a Certificate Payload or can be generated by enrolling the mobile device with Simple Certificate Enrollment Protocol (SCEP).

The MDM Agent communications uses the iOS Security Framework as described in section 3.3.2 TLS Configuration. Configuring the device’s TLS protocol automatically configures the MDM Agent communications. If an additional CA certificate needs to be added to support the MDM Server, see section 3.3.2.3.

2.3.4 Device Un-enrollment Prevention

During the enrollment process, a Configuration Profile including an MDM Payload is loaded onto the mobile device and used to associate the mobile device to an MDM Server. If the MDM Payload is removed, the mobile device will no longer be enrolled with the MDM server and can no longer be considered to be in the evaluated configuration.

As described in [IOS_CFG], the mobile device administrator can specify the *PayloadRemovalDisallowed* key to allow or disallow the ability of a mobile device user to remove the MDM Payload from the device.

The mobile device must be in Supervised Mode to lock the MDM Payload to the device.

An MDM Payload can have a removal password associated with it. If the *PayloadRemovalDisallowed* key is set to prevent unenrollment and the MDM Payload has a removal password associated with it, the mobile device user can unenroll the mobile device only if the mobile device user knows the removal password.

2.3.5 MDM Agent Alerts

The iOS MDM Agent generates and sends an alert in response to an MDM server request for applying a Configuration Profile and in response to receiving a reachability event. These responses are always enabled.

When the application of a Configuration Profile to a mobile device is successful, the MDM Agent replies with an MDM Result Payload with Status value "Acknowledged".

When the application of a Configuration Profile is unsuccessful, the MDM Agent replies with an MDM Result Payload with *Status* value "Error" or *CommandFormatError*, "Idle" and "NotNow".

When a reachability event is received by the iOS MDM Agent, the MDM Agent replies with an MDM Result Payload to acknowledge that the mobile device received the event.

More information on the MDM Result Payload is found in [IOS_MDM].

2.3.6 The MDM Payload

The Mobile Device Management (MDM) Payload, a simple property list, is designated by the com.apple.mdm value in the PayloadType field.

Payload	Key	Setting
MDM	*PayloadRemovalDisallowed*	Must be set to true
MDM	*AccessRights*	Must be set to a value that includes the logical OR with “8”.

Table 5: Essential MDM Payload keys for the evaluated configuration

3 Mobile Device Configuration

This section provides more detailed guidance to configure the supervised mobile devices in the way that conforms to the requirements of the CC evaluation.

This section provides details of the dictionary key values that must be used, or where certain options for the key value are not allowed, in order to meet the requirements of the evaluated configuration described in the [ST].

For dictionary keys not mentioned in this document, please refer to the deploying organization's security policies.

3.1 General Restrictions

3.1.1 Keys for General Restrictions

Payload	Key	Description
Restrictions	*allowAssistant*	Must be set to 'false'. (Siri is not allowed in the evaluated configuration.)
Restrictions	*allowAssistantUserGeneratedContent*	Must be set to 'false'. (Siri is not allowed in the evaluated configuration.)
Restrictions	*allowAssistantWhileLocked*	Must be set to 'false'. (Siri is not allowed in the evaluated configuration.)
Restrictions	*allowLockScreenControlCenter*	Must be set to 'false'.
Restrictions	*allowEnablingRestrictions*	Must be set to 'false'.
Restrictions	*allowUSBRestrictedMode*	Must be set to 'true'.

Table 6: Essential keys in the Restrictions Payload

3.2 Cryptographic Support Functions

The mobile devices include three cryptographic modules that provide the cryptographic support used by the mobile devices.

- Apple CoreCrypto Cryptographic Module for ARM, v9.0 (User Space)
- Apple CoreCrypto Kernel Cryptographic Module for ARM, v9.0 (Kernel Space)
- Apple Secure Key Store Cryptographic Module, v9.0

Warning: The use of other cryptographic engines beyond those listed above was neither evaluated nor tested during the mobile device's Common Criteria evaluation.

The approved mode of operation for these cryptographic modules is configured by default and cannot be changed by the mobile device user or administrator. If the mobile device starts up successfully, then the modules have passed all self-tests and are operating in the approved mode.

3.2.1 Key Generation, Signature Generation and Verification

3.2.1.1 General information

The mobile devices generate the following asymmetric keys.

- Rivest-Shamir-Adleman (RSA) with key sizes of 2048 bits or greater
- Elliptic-curve cryptography (ECC) with NIST curves P-256 and P-384, with key sizes of 256 bits and 384 bits respectively
- ECC curve 25519, with a key size of 256 bits
- Finite-field cryptography (FFC) with key sizes of 2048 bits or greater

3.2.1.2 Mobile device users

For the evaluated configuration, no configuration is required from the mobile device user.

3.2.1.3 Mobile device administrators

For the evaluated configuration, no configuration is required from the mobile device administrator.

3.2.2 Key Establishment

3.2.2.1 General information

The mobile devices use the following for key establishment.

- RSA-based scheme
- ECC-based scheme
- Diffie-Hellman (DH)-based scheme

Key establishment is used for TLS and IKE.

3.2.2.2 Mobile device users

For the evaluated configuration, no configuration is required from the mobile device user.

3.2.2.3 Mobile device administrators

For the evaluated configuration, no configuration is required from the mobile device administrator.

3.2.3 Hashing

3.2.3.1 General information

The mobile devices perform the hash functions secure hash algorithm (SHA)-1, SHA-256, SHA-384, and SHA-512 with message digest sizes 160, 256, 384, and 512 bits.

Functions to perform hashing are provided as part of the Apple CoreCrypto libraries. The invoking function dictates which SHA function is used. Neither the mobile device user nor the mobile device administrator has the ability to configure this choice.

Similarly, each TLS ciphersuite uses a specific and appropriate SHA function. Neither the mobile device user nor the mobile device administrator has the ability to configure this choice.

3.2.3.2 Mobile device users

For the evaluated configuration, no configuration is required from the mobile device user.

3.2.3.3 Mobile device administrators

For VPN connections with IKEv2, the integrity algorithm to be used is selectable by the mobile device administrator by setting the *IntegrityAlgorithm* key. Note that setting *IntegrityAlgorithm* to 'SHA1-96' is not allowed in the evaluated configuration.

3.2.4 Random Number Generation

3.2.4.1 General information

For random bit generation, the mobile devices use a deterministic random bit generator (DRBG), seeded by an entropy source. That source accumulates entropy from software-based noise, and seeds the DRBG with a minimum of 256 bits of entropy.

3.2.4.2 Mobile device users

For the evaluated configuration, no configuration is required from the mobile device user.

3.2.4.3 Mobile device administrators

For the evaluated configuration, no configuration is required from the mobile device administrator.

3.2.5 Keys/Secrets Import/Destruction

3.2.5.1 General information

Cryptographic keys are stored in keychains. In iOS, an application only has access to its own keychain items, so access restrictions are automatically satisfied.

The "Keychain Services Programming Guide" [KEYCHAINPG] describes how keychain items are created, managed, and deleted.

3.2.5.2 Mobile device users

For the evaluated configuration, no configuration is required from the mobile device user.

3.2.5.3 Mobile device administrators

For the evaluated configuration, no configuration is required from the mobile device administrator.

3.2.6 Keys for Configuring Cryptographic Functions

This section provides details of dictionary key values which must be used or which are not allowed to be used in order to meet the requirements of the evaluated configuration described in the [ST].

Payload	Key	Description
VPN	*EncryptionAlgorithm*	May be set to one of the following: • 'AES-128' • 'AES-256' (Default) • 'AES-128-GCM' (16-octet ICV) • 'AES-256-GCM' (16-octet ICV) 'DES' and '3DES' are not allowed in the evaluated configuration. Note that 'AES-128' and 'AES-256' use the CBC mode of operation.
VPN	*IntegrityAlgorithm*	May be set to one of the following: • 'SHA1-160' • 'SHA2-256' (Default) • 'SHA2-384' • 'SHA2-512' 'SHA1-96' is not allowed in the evaluated configuration.
VPN	*DiffieHellmanGroup*	Set to one of the following: '5', '14', '15', '19', or '20'.

Table 7: Essential keys for Configuring Cryptographic Functions

3.3 Network Protocols

3.3.1 EAP-TLS Configuration

3.3.1.1 General information

For Extensible Authentication Protocol (EAP)-TLS, iOS implements TLS 1.0, TLS 1.1, and TLS 1.2 supporting the cipher suites listed in Table 8: EAP-TLS Ciphersuites.

In the evaluated configuration, the mobile devices must use only the EAP-TLS cipher suites.

Ciphersuite Name
TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256

Table 8: EAP-TLS Ciphersuites

No additional configuration is needed for the automatic recovery of a broken Wi-Fi connection.

3.3.1.2 Mobile device user

For the evaluated configuration, no configuration is required from the mobile device user.

3.3.1.3 Mobile device administrator

The cipher suites in Table 8: EAP-TLS Ciphersuites above are automatically selected by the mobile devices (i.e., the mobile devices do not support the individual selection of EAP-TLS cipher suites) when Wi-Fi Protected Access (WPA)-EAP is configured via Configuration Profile as follows.

- *EncryptionType* key must be set to 'WPA2'.
- *AcceptEAPTypes* key must be set to '13', the value representing EAP-TLS.

Because the evaluation of the mobile devices included TLS versions 1.0, 1.1, and 1.2, setting the *TLSMinimumVersion* and *TLSMaximumVersion* keys is a matter for the deploying organization's policy. These keys configure the minimum and maximum TLS versions to be used with EAP-TLS authentication. The default minimum value is '1.0' and the default maximum value is '1.2'.

3.3.2 TLS Configuration

3.3.2.1 General information

TLS is provided by the APIs of the iOS Security Framework, which uses the Apple CoreCrypto Cryptographic Module for ARM, v9.0.

The library implements TLS 1.0, 1.1, and 1.2 supporting the cipher suites listed in Table 9: TLS Ciphersuites. In the evaluated configuration, only TLS 1.2 is supported. The [ST] limits the cipher suites used by TLS connections in the evaluated configuration.

The supported cipher suites below are automatically selected by the mobile devices (i.e., the devices do not support the individual selection of TLS cipher suites). The TLS cipher suites available are defined by the TLS server where all cipher suites listed in the [ST] are always available. Thus, no additional configuration is required by the administrator.

Ciphersuite Name
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

Table 9: TLS Ciphersuites

There are some trusted root certificates that are preinstalled with iOS in a Trust Anchor Database to establish a chain of trust. These certificates are automatically trusted, and

do not need to be included when creating a Configuration Profile. A list of iOS trusted root certificates can be found at [TRUST_STORE].

There are also blocked and always-ask certificates in the Trust Anchor Database. Blocked certificates are believed to be compromised and are never trusted. Always-ask certificates prompt the user whether they want to trust the certificate. Lists of these certificates can also be found at [TRUST_STORE].

3.3.2.2 Mobile device users

For the evaluated configuration, no configuration is required from the mobile device user.

3.3.2.3 Mobile device administrators

TLS/HTTPS Configuration

The mobile device must be configured to automatically reject untrusted HTTPS certificates rather than prompting the user to ask whether to accept it. This is done by setting the *allowUntrustedTLSPrompt* key to 'false'.

Some restrictions must be placed on AirPrint to ensure that it both only uses TLS, and only uses trusted certificates for TLS communication. This is done by setting both the *ForceTLS* key in the AirPrint payload and the *forceAirPrintTrustedTLSRequirement* key in the Restrictions payload to 'true'.

The mobile device administrator must also configure the *TLSTrustedServerNames* and *PayloadCertificateAnchorUUID* keys such that they specify which server certificate common names and certificates will be accepted by the mobile device.

Reference Identifier Configuration

Guidance documentation for setting the reference identifier for certification validation in TLS is provided in the "Obtain policies for establishing trust" section of the "Policies" chapter in [CKTSREF].

Certificate Authority (CA) Configuration

Additional CAs can be added to the mobile device by using a Configuration Profile with the *EAPClientConfiguration, PayloadCertificateArchorUUID,* and *TLSTrustedServerNames* keys.

Client Certificate Configuration

A client certificate with its keys can be installed on the mobile device using a Certificate payload in the Configuration Profile, as described in [IOS_CFG].

Configuration of the Supported Elliptic Curves Extension

The supported elliptic curves below are automatically selected by the mobile devices (i.e., the mobile devices do not support the individual selection of elliptic curves). The [ST] limits the curves used by TLS connections in the evaluated configuration. The curves available are defined by the server where all curves listed in the [ST] are always available. This behavior does not require any additional configuration by the mobile device administrator.

The following curves are available.

- secp256r1
- secp384r1

Curve x25519 is also supported by the mobile devices, and may be disabled in the operational environment.

3.3.3 IPsec Configuration

3.3.3.1 General information

The mobile devices implement IPsec natively, as part of their operating system, so any processing of packets used in IPsec communication takes place on the mobile device. IPsec VPN tunnels are configured and controlled by the Network Extension Framework, which is a part of the Core OS Layer of the mobile devices' operating system.

The Security Policy Database (SPD) is created and configured by defining exceptions for IP traffic routing in a Configuration Profile. By default, all IP traffic is sent through a protected channel between the devices and the desired endpoint (PROTECT in the SPD). Any deviations from the default routing behavior must be explicitly specified as exceptions in the Configuration Profile.

Packet processing exceptions can be created for applications which make use of Captive Networking Identifiers (Captive Networking Apps), as well as for VoiceMail, AirPrint, and CellularServices. The mobile device administrator will need to refer to their organization's security policies to determine whether exceptions should be created and how those exceptions should be configured.

Exceptions for Captive Networking Apps can be configured to allow traffic for these apps to pass outside the tunnel (BYPASS in the SPD). Exceptions for VoiceMail, AirPrint, and CellularServices can allow traffic to pass unencrypted outside the tunnel (BYPASS in the SPD) or drop the traffic entirely (DISCARD in the SPD).

When the VPN is configured as Always-On, the mobile device uses IKEv2 for security association (SA) establishment. Since the mobile device must be configured with Always-On VPN in order to be in the evaluated configuration, the use of IKEv2 does not need to be configured separately.

3.3.3.2 Mobile device users

For the evaluated configuration, no configuration is required from the mobile device user.

3.3.3.3 Mobile device administrators

To configure exceptions for VoiceMail, AirPrint, and CellularServices, the mobile device administrator can specify a *ServiceExceptions* array in the AlwaysOn dictionary of the VPN payload.

Each entry in a *ServiceExceptions* array lists a *ServiceName* key and a corresponding *Action* key. The allowed values for *ServiceName* and *Action* can be found in Table 11: Essential Keys for the VPN Payload. For each *ServiceName*, the corresponding *Action* can be set to 'Allow' (BYPASS in the SPD) or 'Drop' (DISCARD in the SPD).

To configure exceptions for Captive Networking Apps, the mobile device administrator can use the *AllowCaptiveWebSheet*, *AllowAllCaptiveNetworkPlugins*, and *AllowedCaptiveNetworkPlugins* keys in the Configuration Profile. Information on these keys can be found in [IOS_CFG].

When the VPNType key is set to 'AlwaysOn', a catch-all PROTECT rule is created in the SPD. Any traffic not covered by an exception will be covered by that rule.

The mobile device administrator must not declare conflicting traffic exceptions, e.g. declaring both an ‘Allow’ and a ‘Drop’ value for ‘VoiceMail’. This guarantees that the SPD is unambiguous and unaffected by the ordering of SPD entries.

3.3.4 Bluetooth Configuration

3.3.4.1 General information

On iOS, manual authorization for Bluetooth connections is implicitly configured, as Bluetooth pairing can only occur when the mobile device is explicitly made discoverable through the *Settings » Bluetooth* interface. When the mobile device is made discoverable in this manner, another device (or the mobile device itself) can send a pairing request. Commonly, a six-digit number is displayed on both sides which must be manually matched by a mobile device user, i.e. the PIN is shown and the user must accept it before the pairing will complete. If one device does not support this automatic exchange of a PIN, a window for entering a manual PIN is presented to the user. The PIN entered must match on both sides.

Two conditions must be met for the mobile device to become discoverable: Bluetooth must be enabled, and the Bluetooth configuration panel must be both active and in the foreground. If the Bluetooth configuration panel is not the active panel, or if Bluetooth is disabled, the mobile device is not discoverable. There is no other method to make the mobile device discoverable or not discoverable.

Devices that want to pair with the evaluated devices via Bluetooth are required by iOS to use Secure Simple Pairing, which uses Elliptic Curve Diffie-Hellman (ECDH) based authentication and key exchange.

iOS requires that remote Bluetooth devices use an encrypted connection. Connections via Bluetooth/LE are secured using AES-128 in CCM mode. Further information about Bluetooth security is found in [BT]. This behavior requires no additional configuration by the mobile device administrator.

3.3.4.2 Mobile device users

For instructions on how to turn Bluetooth on and off and how to pair and unpair a Bluetooth device, the mobile device user can refer to “Connect Bluetooth devices” in either the “Use iPhone with other devices” section of the [iPhone_UG] or the “Use iPad with other devices” section of the [iPad_UG] “iPhone and other devices” section “Connect Bluetooth devices”. Bluetooth can be disassociated by the mobile device user via the Control Center.

3.3.4.3 Mobile device administrators

In the evaluated configuration, the mobile device administrator can allow or disallow the mobile device user from making modifications to Bluetooth settings on the mobile device by using the *allowBluetoothModification* key in a Configuration Profile.

3.3.5 VPN Configuration

3.3.5.1 General information

In the evaluated configuration, the VPN must be in its Always-On configuration. The Always-On VPN configuration enables the organization to have full control over supervised device traffic by tunneling all IP traffic back to the organization.

3.3.5.2 Mobile device users

For the evaluated configuration, no configuration is required from the mobile device user.

3.3.5.3 Mobile device administrators

The mobile device administrator uses the VPN Policy Payload to configure a traditional systemwide VPN based on IPsec, to specify Internet Key Exchange Version 2 (IKEv2) settings, and to specify attributes such as:

- the Always-On VPN configuration,
- the Certificate authentication method, and
- administrator-defined certificates.

Always-On VPN must be enabled by setting the *VPNType* key to 'AlwaysOn' in the Configuration Profile. When 'AlwaysOn' is selected as the *VPNType* for a Configuration Profile, the corresponding *ProtocolType* key must be set to 'IKEv2'. The *Interfaces* array, which lists the interfaces a particular Always-On VPN configuration applies to, can optionally be specified as 'Cellular, WiFi' (Default), 'Cellular', or 'WiFi'.

IKEv2 must be configured using the IKEv2 Dictionary Keys. The mobile device administrator must specify the IP address or hostname of the VPN server via *RemoteAddress*, the client identifier via *LocalIdentifier*, the remote identifier via *RemoteIdentifier*, the authentication method as 'Certificate' via *AuthenticationMethod*, and the certificate to be used for authentication via *PayloadCertificateUUID*.

Optional keys can be configured which allow:

- enabling extended authentication via *ExtendedAuthEnabled*;
- the specification of a username and password via *AuthName* and *AuthPassword*;
- the specification of the interval the connection is kept alive when the peer cannot be reached via *DeadPeerDetectionRate*;
- the specification of the Common Name of the server certificate issuer and/or the Common Name of their server certificate via *ServerCertificateIssuerCommonName* and *ServerCertificateCommonName*; and
- the specification of *IKESecurityAssociationParameters* and *ChildSecurityAssociationParameters*, both of which allow the further specification of an *EncryptionAlgorithm*, an *IntegrityAlgorithm*, and a *DiffieHellmanGroup* as described in Table 12: Essential keys for Data Protection.

3.3.6 Keys for Configuring Network Protocols

This section provides details of the dictionary key values that must or must not be used in order to meet the requirements of the evaluated configuration described in the [ST].

For dictionary keys not mentioned in this document, please refer to the deploying organization's security policies.

3.3.6.1 TLS Configuration Keys

Payload	Key	Description
Restrictions	*allowUntrustedTLSPrompt*	Must be set to 'false'.
Restrictions	*forceAirPrintTrustedTLSRequirement*	Must be set to 'true'.
AirPrint	*ForceTLS*	Must be set to 'true'.
Wi-Fi	*EncryptionType*	Must be set to 'WPA2'.
	EAPClientConfiguration Dictionary Keys	
Wi-Fi	*AcceptEAPTypes*	Must be set to '13' (EAP-TLS).
Wi-Fi	*PayloadCertificateAnchorUUID*	Must contain at least one UUID of a certificate payload that is to be trusted. Note that setting this key prevents the mobile device from asking the user if certificates are trusted.
Wi-Fi	*TLSTrustedServerNames*	Must be set.
Wi-Fi	*TLSCertificateIsRequired*	Must be set to 'true'.

Table 10: Essential Payload Keys for TLS and EAP-TLS

3.3.6.2 VPN Configuration keys

Payload	Key	Description
VPN	*VPNType*	Must be set to 'AlwaysOn'.
VPN	*OnDemandEnabled*	Must be set to '0'.
	IKEv2 Dictionary Keys	
VPN	*RemoteAddress*	Must be set. Specifies the IP address or hostname of your organization's VPN server.
VPN	*LocalIdentifier*	Must be set.
VPN	*RemoteIdentifier*	Must be set.
VPN	*AuthenticationMethod*	Must be set to 'Certificate'.
VPN	*PayloadCertificateUUID*	Must be set. Specifies the universally unique identifier (UUID) of the identity certificate used as the account credential.

VPN	*CertificateType*	Must be set to one of the following: • RSA (Default) • ECDSA256 • ECDSA384 Specifies the type of PayloadCertificateUUID used for IKEv2 machine authentication.
VPN	*ServerCertificateIssuerCommonName*	Must be set. Specifies the Common Name of the server certificate issuer. This key will cause IKE to send a certificate request to the server based on the specified certificate issuer.
VPN	*EnableCertificateRevocationCheck*	Must be set to '1'. Enables a certificate revocation check for IKEv2 connections.
VPN	*IKESecurityAssociationParameters*	Optional. A dictionary which specifies the parameters for IKEv2 IKE_SA_INIT and IKE_AUTH exchanges (Phase 1).
VPN	*ChildSecurityAssociationParameters*	Optional. A dictionary which specifies the parameters for IKEv2 child SAs (Phase 2). If parameters are not specified for Phase 2, the Phase 1 parameters will be used. If the corresponding Phase 1 parameters are also not specified, the default values for those parameters will be used.
	IKESecurityAssociationParameters and ChildSecurityAssociationParameters Dictionary Keys	
VPN	*EncryptionAlgorithm*	May be set to one of the following. • 'AES-128' • 'AES-256' (Default) • 'AES-128-GCM' (16-octet ICV) • 'AES-256-GCM' (16-octet ICV) 'DES' and '3DES' are not allowed in the evaluated configuration. Note that 'AES-128' and 'AES-256' use the CBC mode of operation.

VPN	*IntegrityAlgorithm*	May be set to one of the following. • 'SHA1-160' • 'SHA2-256' (Default) • 'SHA2-384' • 'SHA2-512' 'SHA1-96' is not allowed in the evaluated configuration.	
VPN	*DiffieHellmanGroup*	Set to one of the following: '5', '14', '15', '19', or '20'.	
VPN	*LifeTimeInMinutes*	Optional. SA lifetime (rekey interval) in minutes. Allowed values are '10' through '1440'. Defaults to '1440' (24 hours).	
	AlwaysOn Dictionary Keys		
VPN	*UIToggleEnabled*	Optional. If set to '1', allows the mobile device user to disable this VPN configuration. Defaults to '0'.	
VPN	*TunnelConfigurations*	*ProtocolType*	Must be set to 'IKEv2'
		Interfaces	Optional. An array which lists the interfaces to which this configuration applies. Valid array entries are 'Cellular' and 'WiFi'. Defaults to 'Cellular, WiFi'.
VPN	*ServiceExceptions*	*ServiceName*	The name of a system service which is exempt from AlwaysOn VPN. May be set to one of the following. • VoiceMail • AirPrint • CellularServices
		Action	May be set to one of the following. • Allow • Drop
VPN	*AllowCaptiveWebSheet*	Optional. If set to '1', allows traffic from Captive Web Sheet outside the VPN tunnel. Defaults to '0'.	
VPN	*AllowAllCaptiveNetworkPlugins*	Optional. If set to '1', allows traffic from all Captive Networking apps outside the VPN tunnel to perform Captive network handling. Defaults to '0'.	

VPN	*AllowedCaptiveNetworkPlugins*	Optional. An array of dictionaries which describes Captive Networking apps whose traffic will be allowed outside the VPN tunnel to perform Captive network handling. Used only when *AllowAllCaptiveNetworkPlugins* is '0'. Each dictionary in this array must contain a *BundleIdentifier* key of type string, the value of which must be the application's bundle identifier.
	DNS Dictionary Keys	
VPN	*SupplementalMatchDomains*	Must not be set. (This key is used to create a split DNS, which is not allowed in the evaluated configuration.)

Table 11: Essential Keys for the VPN Payload

3.4 Data Protection

3.4.1 Data-At-Rest (DAR) Protection Configuration

3.4.1.1 General information

To ensure data at rest protection, establishment of a passcode on the mobile device is required.

3.4.1.2 Mobile device users

Users can check that data at rest protection is enabled on their device at *Privacy and security » Security » Security*, looking at the passcode settings screen, and also by seeing that a passcode is required to access the device. No further configuration is required.

3.4.1.3 Mobile device administrators

Mobile device administrators must ensure that mobile device users set a passcode by using the *forcePin* key in the Passcode Policy Payload. Other keys available in this payload allow administrators to configure passcode requirements to their deploying organizations policy.

See 3.5.1, Passcode Authentication Configuration, for more information on passcode configuration.

3.4.2 Restrict Application Access to System Services

3.4.2.1 General information

Access control to system services in the Core Services layer is hardcoded and thus not configurable by the mobile device user or administrator.

Access control for applications to system services can be restricted on a per-app basis. In iOS 12, these services are as follows.

- Location Services
- Contacts
- Calendars
- Reminders
- Photos
- Bluetooth Sharing
- Microphone
- Speech Recognition
- Camera
- Health
- HomeKit
- Media & Apple Music
- Motion & Fitness

3.4.2.2 Mobile device users

A list of system services can be obtained from the mobile device *Settings » Privacy*. For each system service, the Applications which have permission to use that service can be inspected and changed.

3.4.2.3 Mobile device administrators

Mobile device administrators can not specify access control for applications to system services.

3.4.3 Wiping of Protected Data

3.4.3.1 General information

A wipe operation is performed after the mobile device user exceeds the limit number of failed authentication attempts or upon receiving a request from an authorized administrator. The administrator can configure the number of failed attempts by using the following Configuration Profile key in the Passcode Policy Payload: *maxFailedAttempts*. This key takes an integer value between ‘2’ and ‘11’.

3.4.3.2 Mobile device users

The mobile device user can wipe the device themself. This is described in [iPhone_UG] and [iPad_UG] in “*Restart, update, reset, and restore”*, and is accessed on the device from *Settings » General » Reset » Erase all Content and Settings.* Depending on the organizational policy, the mobile device administrator can disable this function.

3.4.3.3 Mobile device administrators

It is mandatory that the mobile device administrator can issue a remote wipe command from the MDM server using the MDM protocol as described in [IOS_MDM].

The following key is required to execute a remote device wipe: *RequestType* with a value of 'EraseDevice'. Upon receiving this command, the device immediately erases itself. No warning is given to the user. This command is performed immediately even if the device is locked.

In order to execute this command successfully, Device Erase access rights must be set. To enable this access, the following MDM Payload related key must be used: *AccessRights*. The value for this key is determined by a logical "OR" that includes the value '8', where 8 stands for allowing device erase rights.

Depending on the organizational policy, the mobile device administrator can disallow the mobile device user from wiping the device themself. This ability can be configured by the mobile device administrator by setting the *allowEraseContentAndSettings* key to 'false'.

3.4.4 Keys for Configuring Data Protection

This section provides details of the dictionary key values that must be used, or where certain options for the key value are not allowed, in order to meet the requirements of the evaluated configuration described in the [ST].

For dictionary keys not mentioned in this document, please refer to the deploying organization's security policies.

Payload	Key	Description
MDM	*RequestType*	EraseDevice (**Warning:** Only set this if the intention is to wipe the mobile device.)
MDM	*AccessRights*	A logical "OR" including the value "8"
Passcode Policy	*MaxFailedAttempts*	A value between '2' and '11' according to the organizations security policy
Restrictions	*allowEraseContentAndSettings*	Disables the option to erase all content and settings from the mobile device UI

Table 12: Essential keys for Data Protection

3.5 Identification & Authentication

3.5.1 Passcode Authentication Configuration

3.5.1.1 General information

In the evaluated configuration, mobile devices must be configured to use either a numeric passcode or an alphanumeric passcode.

The Passcode Policy Payload is described in [IOS-CFG] and describes the keys that can be used to set attributes such as:

- defining the minimum passcode length,
- defining requirements for the passcode complexity,
- defining the maximum passcode lifetime,

- defining the maximum time of inactivity after which the mobile device is locked automatically, and
- defining the maximum number of consecutive authentication failures after which the mobile device is wiped.

The devices allow the following parameters for passcode complexity.

- Passcodes can be composed of be composed of any combination of upper and lower case letters, numbers, and special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”
- Passcode length must be between 1 and 16.

3.5.1.2 Mobile device users

In the evaluated configuration, the mobile device user cannot configure the passcode policy.

3.5.1.3 Mobile device administrators

It is mandatory that the mobile device administrator configures the passcode policy for the mobile device.

The Passcode Policy Payload presents the administrator with an alphanumeric passcode entry mechanism, which allows for the entry of arbitrarily long and complex passcodes including the selection of special characters. To do this, set the configuration keys *allowSimple* to ‘false’ and *RequireAlphanumberic* to ‘Yes’.

Also, set the configuration key *minLength* to a value greater than zero, defined by the deploying organization's policy.

3.5.2 Protected Authentication Feedback

3.5.2.1 General information

All passcode entries are obscured by iOS. This is done by displaying a dot symbol in place of each character as the passcode entry user input occurs. No configuration of this feature is required from the mobile device administrator.

Biometric authentication inputs do not provide feedback to the user unless the input is rejected. Additionally, biometric authentication inputs do not relay authentication entry information and are inherently obscured. When an invalid fingerprint sample is given or a fingerprint sample cannot be authenticated, a simple error message is returned which prompts the user to try again. When an invalid facial sample is given or a facial sample cannot be authenticated, the mobile device will vibrate. If three invalid biometric samples are presented the mobile device will offer passcode entry. After five invalid biometric samples are presented passcode authentication is required.

Refer to [PASSCODE_Help] for more information on how to manage a passcode.

3.5.2.2 Mobile device users

Passcode entry is obscured by iOS, no configuration of this feature is required from the mobile device user.

3.5.2.3 Mobile device administrators

Passcode entry is obscured by iOS, no configuration of this feature is required from the mobile device administrator.

3.5.3 Biometric Authentication Factors

3.5.3.1 General information

Enrollment and management of biometric authentication factors and credentials is detailed in [iPhone_UG] and [iPad_UG], and found on the device at *Settings » Face ID & Passcode* or *Settings » Touch ID & Passcode*.

3.5.3.2 Mobile device users

In the evaluated configuration, the mobile device user cannot enable Touch ID or Face ID. Only the mobile device administrator can enable/disable Touch ID or Face ID using the Restrictions Payload. If the mobile device administrator has enabled these biometric authentication factors, the following is guidance on how the mobile user can configure Touch ID and Face ID.

Enrollment for Touch ID is typically accomplished during initial device configuration but can also be performed using the *Settings »Touch ID & Passcode* menus. Multiple fingerprints may be enrolled, named, and deleted from this menu. In order to remove a specific finger, a device user must tap the finger for removal followed by delete fingerprint. Mobile device users may place a finger on the Touch ID sensor to determine which biometric credential entry it is mapped to. Users may also disable Touch ID selectively for applications, or entirely, from the *Settings » Touch ID & Passcode* menu, by authenticating using their passcode and turning off one or more of the following corresponding options.

- Unlock
- Apple Pay
- iTunes & App Store

Enrollment for Face ID is typically accomplished during initial device configuration but can also be performed using the *Settings » Face ID & Passcode* menu by tapping the “Set up Face ID” option. Mobile users can enroll an alternative appearance for Face ID, for a total of two enrollments per device. Mobile users may establish Face ID credentials by providing biometric samples. They may also remove biometric samples from the *Settings » Face ID & Passcode* menu by tapping the *Reset Face ID* option. This action removes all established Face ID credentials. Users may also disable Face ID selectively for applications, or entirely, from the *Settings » Face ID & Passcode* menu by turning off one or more of the following corresponding options.

- Unlock
- Apple Pay
- iTunes & App Store
- Safari AutoFill

3.5.3.3 Mobile device administrators

A mobile device administrator can configure to not allow a device user to enable Touch ID or Face ID by setting the key *AllowFingerprintForUnlock* to false in a Configuration Profile using the Restrictions Payload.

3.5.4 Authentication Attempt Configuration

3.5.4.1 General information

Both Face ID and Touch ID allow up to five unsuccessful authentication attempts before passcode authentication is required. For the details, please see section "Touch ID, Face ID, and passcodes" in [IOS_SEC].

3.5.4.2 Mobile device users

In the evaluated configuration, the mobile device user cannot configure the maximum number of failed authentication attempts.

3.5.4.3 Mobile device administrators

To limit/configure the number of consecutive failed authentication attempts for the passcode; the administrator can use the key *maxFailedAttempts* This key takes an integer value between '2' and '11'. See the Passcode Policy Payload in section 3.5.1, Passcode Authentication Configuration.

3.5.5 Re-Authentication Configuration

3.5.5.1 General information

When the use of a passcode is enabled, the mobile device automatically prompts the user for a passcode to unlock the device. No additional configuration is required.

Use of Touch ID or Face ID can be set in the *Settings » Touch ID & Passcode* or *Settings » Face ID & Passcode.* The biometric authentication factor can be configured for device unlock, Apple Pay and iTunes and App Store.

The Passcode Policy Payload allows an administrator to enable/disable modification of Touch ID or Face ID through specification of the *allowFingerprintModification* key.

A passcode must be supplied for additional security validation in any of the following instances.

For Touch ID

- The mobile device has just been turned on or restarted
- For device software updates
- To wipe the device
- To view or change passcode settings
- To install iOS Configuration Profiles

For Face ID

- The mobile device has just been turned on or restarted
- The mobile device hasn't been unlocked for more than 48 hours

- The passcode hasn't been used to unlock the mobile device in the last six and a half days and Face ID hasn't unlocked the mobile device in the last 4 hours
- The mobile device has received a remote lock command
- After five unsuccessful attempts to match a face
- After initiating power off/Emergency SOS by pressing and holding either volume button and the side button simultaneously for 2 seconds

3.5.5.2 Mobile device users

In the evaluated configuration, the mobile device user cannot enable/disable the modification of Touch ID or Face ID.

3.5.5.3 Mobile device administrators

In the evaluated configuration, the mobile device administrator set the *allowFingerprintModification* key to a value defined by the organization's policy.

3.5.6 X.509 Certificate Configuration

3.5.6.1 General

X.509 certificates are configured by an administrator using the keys of the *Certificate Payload* in a Configuration Profile; see [IOS_CFG].

Certificates have a certificate type that defines their respective application area. This ensures that only certificates defined for a specific application area are used. In addition, the database containing trust anchors for all certificates is protected via integrity check and write protection. The certificate types supported by the devices are as follows.

- AppleX509Basic
- AppleSSL
- AppleSMIME
- AppleEAP
- AppleIPsec
- AppleCodeSigning
- AppleIDValidation
- AppleTimeStamping

The mobile devices have a Trust Anchor Database which contains trusted root certificates preinstalled with iOS; see [TRUST_STORE]. These preinstalled trusted root certificates cannot be modified. New certificates can be added to the Trust Anchor Database or currently installed certificates can be removed.

3.5.6.2 Mobile device users

In the evaluated configuration, the mobile device user cannot import X.509v3 certificates into the Trust Anchor Database. However, if the mobile device is unsupervised, the mobile device user can install root certificates into the Trust Anchor Database.

Unless the administrator has disallowed the removal of the Configuration Profile that contains the certificate, mobile device users can manually remove certificates that have been installed on their device. Choose *Settings » General » Profile & Device Management » Profiles*, select a profile, choose More Details, and then choose the appropriate certificate to remove.

In the evaluated configuration, the mobile device user can remove imported X.509v3 certificates but cannot remove other X.509v3 certificates in the Trust Anchor Database.

3.5.6.3 Mobile device administrators

In the evaluated configuration, mobile device administrators are allowed to modify the Trust Anchor Database. X.509 certificates can be configured by using a configuration profile. Certificate identities can be deployed using the following two methods: 1) using Public Key Cryptography Standards (PKCS) #12 identity certificate and 2) Simple Certificate Enrollment Protocol (SCEP). The mobile device administrator should use the Certificate Payload if using the first option and should use the SCEP Payload if using the second option.

The mobile device administrator can also send the mobile device user an email with the certificate as an attachment or a link to a secure site hosting the certificate. The user will download the certificate, from the email or site, to install on the mobile device.

More information on certificate configuration can be found in [iOSDeployRef] and [IOS_CFG].

In the evaluated configuration, the mobile device administrator must disallow the removal of a Certificate Payload by a user in a Configuration Profile by setting the *PayloadRemovalDisallowed* key for that payload to ‘true’. See the [IOS_CFG] section.

When configuring the devices to utilize EAP-TLS as part of a WPA2 protected Wi-Fi-network, the CA certificate(s) to which the server's certificate must chain can be configured using the *PayLoadCertificateAnchorUUID* key in the Wi-Fi Payload of the Configuration Profile.

Mobile device administrators can view all certificates on a device and remove any certificates it has installed via the MDM protocol using the *RequestType* key with the content “CertificateList”. The MDM protocol also allows for certificate removal.

Certificate Validation

To configure the devices to reject untrusted certificates, the administrator can use the *TLSAllowTrustExceptions* key in the Wi-Fi Payload of the Configuration Profile which enforces that untrusted certificates are not accepted and the authentication fails if such untrusted certificates are presented.

To enforce the verification of the server name defined with the X.509 certificate during the WPA-EAP handshake between the mobile device and the remote access point, the policy must contain the server name to be expected in the certificate with the *TLSTrustedServerNames* key.

Guidance and the API documentation related to certificate validation is provided in the “Certificate, Key, and Trust Services” [CKTSREF] in the section “Trust”. See the function “SecTrustEvaluate”.

3.5.7 Keys for Identification and Authentication

This section provides details of the dictionary key values that must be used, or where certain options for the key value are not allowed, in order to meet the requirements of the evaluated configuration described in the [ST].

Payload	Key	Setting
Passcode Policy	*allowSimple*	Must be set to 'false'.
Passcode Policy	*forcePin*	Must be set to 'true'.
Passcode Policy	*maxFailedAttempts*	Must be set to a value between 2 and 11 according to the deploying organizations policy.
Passcode Policy	*maxInactivity*	Should be set to a value defined by the deploying organization's policy.
Passcode Policy	*maxPINAgeInDays*	Should be set to a value defined by the deploying organization’s policy.
Passcode Policy	*minComplexChars*	Should be set to a value defined by the deploying organization’s policy.
Passcode Policy	*minLength*	Should be set to a value defined by the organization's policy.
Passcode Policy	*requireAlphanumeric*	Should be set to a value defined by the organization's policy.
Passcode Policy	*pinHistory*	Should be set to a value defined by the organization's policy.
Passcode Policy	*maxGracePeriod*	Must be set to 0.
Passcode Policy	*allowFingerprintModification*	Should be set to a value defined by the organization's policy.
Passcode Policy	*changeAtNextAuth*	Should be set to a value defined by the organization's policy.

Table 13: Essential keys for Identification and Authentication

3.6 Security Management

3.6.1 Install/Remove Apps from the Device

3.6.1.1 General information

If the mobile device is enrolled in MDM, managed apps on the mobile device can be removed by an administrator remotely via the MDM System, or when the mobile device user removes their own device from MDM. If a mobile phone is removed from MDM, the

mobile device administrator has some control over what happens to the associated data. When a managed app is removed from a device, the associated data is removed with it.

For more information on managed apps refer to the “iOS Deployment Reference” [iOSDeployRef].

3.6.1.2 Mobile device users

Mobile device users may be able to install or remove an application from their device. (This depends upon the organization’s policy and the value of the dictionary keys in the Restrictions Payload for *allowAppRemoval* and *allowAppInstallation*. See [iPhone_UG] section “Remove apps from iPhone” and the [iPad_UG] section “Remove apps from iPad”.)

3.6.1.3 Mobile device administrators

The mobile device administrator can install applications on the mobile device using an MDM system or Apple Configurator 2. Refer to the [iOSDeployRef] section “App and book distribution”, the [AConfig] section “Add apps”, and the [IOS_MDM] section “InstallApplication Commands Install an Application”. If installing an enterprise application, refer to [IOS_MDM] section “InstallEnterpriseApplication Commands Install an Enterprise Application”.

The mobile device administrator can remove managed applications using MDM. To remove an application, the MDM server sends a command using the *RequestType* and *Identifier* keys. The below table provides additional information these keys.

Key	Description
RequestType	This key must be set to: RemoveApplication
Identifier	The application’s identifier

More information can be found in [IOS_MDM] in a section titled “RemoveApplication Commands Remove Installed Managed Applications”.

3.6.2 Configure Access and Notification in Locked State

3.6.2.1 General information

By default, the following features are available when the mobile device is locked and authentication is not needed:

- Making emergency calls,
- Using the camera, and
- Using the flashlight.

Access to certain optional features can be allowed when the mobile device is in a locked state. These optional features include the following.

- Email notification
- Calendar appointment

- Text message notification

3.6.2.2 Mobile device users

To allow access to the optional features when the mobile device is locked, go to Settings » Touch ID & Passcode (mobile devices with Touch ID) or Settings » FaceID Passcode (mobile devices with Face ID) and select the features you want to allow access under the Allow Access When Locked menu. Those items may be restricted by a Configuration Profile installed by an administrator. Refer to the [iPad_UG] section "Access features from the iPad Lock screen" and the [iPhone_UG] section "Access features from the iPhone Lock screen" for more information.

Certain display notifications can be set when the mobile device is in the locked state. To enable/disable display notifications in the locked state, go to Settings » Face ID & Passcode or Settings » Touch ID & Passcode and enter the passcode. Once authenticated, turn on Notification Center (found below Allow Access When Locked). Refer to the [iPad_UG] section "Change notification settings on iPad" and the [iPhone_UG] section "Change notification settings on iPhone" for more information.

3.6.2.3 Mobile device administrators

The mobile device administrator can use the *allowLockScreenNotification* key in the Restrictions Payload in a Configuration Profile to disallow the user from viewing past notifications (i.e., disable Notification history). However, the mobile device user can see notifications as they arrive. To disable displaying notifications on the lock screen for applications, the *ShowInLockScreen* key in the Notifications Payload must be set to 'true'.

Once the notification settings have been implemented by the mobile device administrator, the *allowNotificationsModification* key in the Restrictions Payload must be set to 'true' if the settings are not allowed to be modified.

Refer to [IOS_CFG] for more information.

3.6.3 Device/Session Locking

3.6.3.1 General information

The mobile device is locked after a configurable time of user inactivity. To unlock the mobile device, an authentication mechanism must be enabled. For example, the device user uses a passcode, possibly with Face ID or Touch ID for authentication.

3.6.3.2 Mobile device users

In the evaluated configuration the mobile device user is not allowed to configure the auto-lock in *Settings » General » Auto-Lock.*

Mobile device users can transition to the locked state by pressing the side button (or for some mobile device models) the Sleep/Wake button.

3.6.3.3 Mobile device administrators

It is mandatory that mobile device administrators configure the device/session locking policy on the mobile devices. This is done by setting the Configuration Profile key *maxInactivity* in the Passcode Policy Payload to the desirable time. The number of

authentication failures allowed is set using the *maxFailedAttempts* key, in the same payload, to a value between '2' and '11'. Refer to [IOS_CFG] for additional information.

Additionally, the mobile device administrator can use the *DeviceLock* key described in [IOS_MDM]. This key requires the Device Lock and Passcode Removal access rights. In the MDM payload, setting the *AccessRights* key to '4' allows for device lock and passcode removal.

3.6.4 Timestamp Configuration

3.6.4.1 General information

In the evaluated configuration, the mobile device must be configured to update its time automatically. Accurate timestamps are crucial when it comes to analyzing audit logs (see Section 4, Security Audit for information on audit logs). The devices can use several time sources to automatically update the time: Network, Identity and Time Zone (NITZ); Global Positioning Satellites (GPS); Network Time Protocol (NTP) standards; or the cellular carrier time service. When configured and maintained using one of these time sources, the time may be considered reliable. Only the NTP is configurable by the mobile device administrator.

3.6.4.2 Mobile device users

In the evaluated configuration, the mobile device user is not allowed to configure the automatic time update options.

3.6.4.3 Mobile device administrators

The mobile device administrator can configure the mobile device to connect to a time server. Using the Time Server Payload, the *timeServer* and *timeZone* keys should be used. The following table provides additional details about these keys.

Key	Description
timeserver	This value represents the network time protocol (NTP) server to connect to.
timeZone	This value represents the timezone. It must be an entry in the `/usr/share/zoneinfo/`. Examples include: "America/Denver" or "Zulu".

The mobile administrator can disallow the mobile user from turning off the "Set Automatically" option for the date and time. In the Restrictions Payload, setting the *forceAutomaticDateAndTime* key to true turns on the Date and Time "Set Automatically" feature and it cannot be turned off by the mobile device user.

Additional information on these settings can be found in [IOS_CFG].

3.6.5 Access Banner Configuration

3.6.5.1 General information

In the evaluated configuration the mobile devices are required to display an access banner as an advisory warning message regarding unauthorized use of the mobile device.

3.6.5.2 Mobile device users

In the evaluated configuration, the mobile device user is not allowed to configure the access banner.

3.6.5.3 Mobile device administrators

Also, the access banner can be configured by creating a background picture with the relevant information and configuring that picture as the background for the lock screen as described in [iOS_MDM] section “Wallpaper Sets the Wallpaper”. This banner is not allowed to be changed by the mobile device user and this can be prevented by specifying the *allowWallpaperModification key* to ‘false’ as described in [IOS_CFG].

The image is sent as a Base64 encoded image (as part of the Wallpaper command). It must be either a PNG or JPEG.

Alternatively, a notice and consent warning message can be configured through an app that provides the requisite notice and acknowledgement functionality rather than through iOS itself. The implementing organization must deploy a customizable application that provides users’ notice of the banner (e.g., through the Apple Push Notification Service) and also the ability to acknowledge the banner content within the application.

3.6.6 Enable/Disable Cameras and Microphones

3.6.6.1 General information

The cameras and microphones on the iPhone and iPad can be managed across the devices or on a per-app basis.

Additional information on these settings can be found in [IOS_CFG].

3.6.6.2 Mobile device users

Mobile device users can optionally disable the use of the cameras on a per-app basis. This can be done on the iPhone or iPad from *Settings » Privacy » Camera*. If the mobile device administrator has restricted the use of the camera then this functionality will not work.

Mobile device users can optionally disable the use of the microphones on a per-app basis. This can be done on the iPhone or iPad from *Settings » Privacy » Microphone*.

3.6.6.3 Mobile device administrators

The mobile device administrator can optionally disallow camera use across the mobile device by using the key *allowCamera* in the Restrictions Payload.

The mobile device administrator can optionally disallow camera use on a per app basis using the key *Camera* in the Privacy Preferences Policy Control Payload.

The mobile device administrator can optionally disallow microphone use on a per app basis using the key *Microphone* in the Privacy Preferences Policy Control Payload.

Refer to [IOS_CFG] for more information.

3.6.7 Enable/Disable Cellular, Wi-Fi, Wi-Fi Hotspot, Bluetooth, NFC

3.6.7.1 General information

The devices contain a variety of radios which can be configured by the users or administrators according to the organization's policy.

3.6.7.2 Mobile device users

Mobile device users can enable/disable cellular by following instructions provided in the [iPhone_UG] and the [iPad_UG]: *Safety, handling, and support » View or change cellular data settings* section.

Mobile device users can enable/disable Bluetooth by following the instructions provided in the [iPhone_UG] the [iPad_UG]: *Use iPhone with other devices » Connect Bluetooth devices to iPhone.*

Mobile device users can enable/disable Wi-Fi by following the instructions provided in the [iPhone_UG] and the [iPad_UG] *Set up and started section » Connect to the Internet.*

Mobile device users can enable/disable Wi-Fi hotspot by following the instructions provided in the [iPhone_UG] and the [iPad_UG]: *Use iPhone with other devices » Share an Internet connection.*

NFC will be disabled if there are no passes stored in the Apple Wallet application. Passes are stored data representing physical cards such as boarding passes and credit cards. When the mobile user adds a pass, NFC is automatically activated. Instructions for adding passes are located in [PAY_SETUP] and removing passes are located in [MANAGE_CARDS].

3.6.7.3 Mobile device administrators

The mobile device administrator can optionally restrict the mobile device from using cellular data by specifying the Network Usage Rules Payload key *AllowCellularData* to 'false'.

The mobile device administrator can optionally restrict the mobile device user from modifying any cellular data settings by using the Restrictions Payload key: *allowAppCellularDataModification*.

The mobile device administrator can optionally enable/disable the ability of the mobile device user to modify Bluetooth settings by using the following Configuration Profile key: *allowBluetoothModification*.

The mobile device administrator can optionally enable/disable Wi-Fi hotspot functionality by using the following key in the WiFi Payload: *IsHotspot*.

Wi-Fi can effectively be enabled/disabled by an administrator setting the Restrictions payload key *forceWiFiWhitelisting*

NFC can be disabled by not having any passes stored in the Apple Wallet application. Passes are stored data representing physical cards such as boarding passes and credit

cards. If there are no passes stored, the mobile device administrator can disable the Wallet application using the Restrictions payload key *blacklistedAppBundleIDs* with a string array containing the value “com.apple.Passbook”. If the Wallet application is not disabled, the mobile device user can add a pass and enable NFC.

Refer to [IOS_CFG] for more information.

3.6.8 Enable/Disable Location Services

3.6.8.1 General information

Additional information on this setting can be found in [IOS_CFG].

3.6.8.2 Mobile device users

Device users can enable/disable location services by following the instructions provided in the [iPhone_UG] and the [iPad_UG]: “Privacy and security” section.

3.6.8.3 Mobile device administrators

The mobile device administrator can enable/disable location services during initial setup of the mobile device. This can occur after a device wipe or setting up the device for the first time. Setting the *skip_setup_items* key to ‘Location’ causes the Setup Assistant to skip the Location Services screens. By skipping these screens, Location Services will not be set up.

More information can be found in [IOS_MDM] and [MDM_SETTINGS_IT].

3.6.9 Secure Software Updates

3.6.9.1 General information

The mobile device startup process helps ensure that only Apple-signed code can be installed on a device. To prevent devices from being downgraded to older versions that lack the latest security updates, iOS uses a process called System Software Authorization. If downgrades were possible, an attacker who gains possession of a device could install an older version of iOS and exploit a vulnerability that has been fixed in the newer version.

Software updates to the mobile devices are released regularly to address emerging security concerns and also provide new features; these updates are provided for all supported devices simultaneously. Mobile device users receive iOS update notifications on the mobile device and also through iTunes. Updates are delivered wirelessly, encouraging rapid adoption of the latest security fixes.

All iOS updates are digitally signed. The user can verify the software version installed on the mobile devices. Refer to section 2.2.2 Verifying the device(s) for more information.

iOS software updates can be installed using iTunes or over-the-air (OTA) on the device. With iTunes, a full copy of iOS is downloaded and installed. OTA software updates download only the components required to complete an update, rather than downloading the entire OS, improving network efficiency. Additionally, software updates can be cached on a local network server running the caching service on OS X Server so that iOS devices do not need to access Apple servers to obtain the necessary update data.

More info about iOS application and system security as well as encryption and data protection can be found in [IOS_SEC].

3.6.9.2 Mobile device users

The integrity and authenticity of software updates is ensured by the design of iOS. There is no configuration for a device user to change that. Mobile device users can Update iOS software on their device. See [iPhone_UG] section "Update iOS software on iPhone" and [iPad_UG] section "Update iOS software on iPad".

3.6.9.3 Mobile device administrators

The integrity and authenticity of software updates is ensured by the design of iOS. There is no configuration for a device administrator to change that. Administrators can delay the availability of OS updates on the device via the Restrictions Payload. User the *forceDelayedSoftwareUpdates* key to enable the feature and the *enforcedSoftwareUpdateDelay* key to define how many days the update should be delayed. The details are described in section of Managed Apps and Updates in [ISO_MDM].

3.6.10 Enable/Disable Remote Backup

3.6.10.1 General information

Backups may be done using iCloud or iTunes. If backup is enabled, iCloud automatically backs up a device daily when the device is connected to power, locked, and on Wi-Fi. Backup can also be done via iTunes by connecting a device to a computer using USB.

3.6.10.2 Mobile device users

Device users can use enable/disable remote backup to iCloud or iTunes by following the instructions provided in the [iPhone_UG] "Back up iPhone using iCloud or iTunes" and the [iPad_UG] "Back up iPad using iCloud or iTunes".

3.6.10.3 Mobile device administrators

In the evaluated configuration, administrators can enable/disable remote backup for the mobile device (e.g., to iCloud, iTunes) using a Configuration Profile. Additional information on these settings can be found in [IOS_CFG].

3.6.11 Configure Application Installation Policy

3.6.11.1 General information

Apple recommends that MDM is used to manage applications for an enterprise. MDM can be used to help users install enterprise apps.

3.6.11.2 Mobile device users

In the evaluated configuration, mobile device users cannot change the application installation policy.

3.6.11.3 Mobile device administrators

It is mandatory that mobile device administrators configure an application installation policy.

This is accomplished by setting *allowAppInstallation* to 'false' in the Restrictions Payload, which means that the App Store is disabled. Mobile device users are unable to install or update their applications.

3.6.12 Importing keys/ shared secrets

3.6.12.1 General information

It is mandatory that key can be imported and destroyed on the mobile devices by the mobile device administrators.

All keys/secrets are automatically stored in secure key storage.

3.6.12.2 Mobile device users

In the evaluated configuration, mobile device users cannot import and destroy keys/secrets.

3.6.12.3 Mobile device administrators

Mobile device administrators can import keys/secrets into the secure key storage by specifying the value when using dictionary keys that are associated with keys/secrets.

3.6.13 Dictionary Keys for Management Functions

Payload	Key	Description
Cameras and Microphones		
Restrictions	*allowCamera*	If set to 'false' will completely disable the cameras.
Privacy Preferences Policy Control	*Camera*	Provide the array of bundle IDs / binary installation path that is not allowed to use the camera.
Privacy Preferences Policy Control	*Microphone*	Provide the array of bundle IDs / binary installation path that is not allowed to use the microphone.
Access Banner		
Restrictions	*allowWallpaperModification*	Must be false.
Date and Time		
Restrictions	*forceAutomaticDateAndTime*	Must be true.

Table 14: Essential keys for Management functions

4 Security Audit

4.1 Audit Logging

iOS logging capabilities collect a wide array of information concerning device usage and configuration. The available commands and responses constitute audit records and must be configured by administrators using Configuration Profiles. The details for profile implementation and audit record collection is located in [IOS_CFG], [IOS_LOGS], and [LOGGING].

Each audit record, at a minimum, contains the following:

- date and time of the event;
- type of event (this is described as log level and log tag)
- subject identity (this is described as PID and PPID)
- the outcome (success or failure) of the event; and
- any applicable required additional information.

Each field of the example log below corresponds with the above format.

Date and Time	Type of event	Subject identity	The outcome
Dec 10 15:22:29.546196	<Error>:	iPadAir2 neagent[446]	Certificate authentication data could not be verified. Failed to process IKE Auth packet.

Figure 1: Example Audit Log

Table *15*: Audit Record Format provides examples of audit events required by [PP_MD_V3.1] as well as the [EP_MDM_AGENT] and [PP_WLAN_CLI_EP].

SFR specified in [ST]	Auditable Events	Additional Audit Record Contents	Example of Audit Records
Device audit records			
FAU_GEN.1(1) {MDF}	Start-up and shutdown of the audit functions	No additional information.	`Dec 5 11:34:58 iPhone-6s mdmd(libdyld.dylib)[6307] <Notice>: mdmd starting…` `Dec 5 11:39:19 iPhone-6s mdmd(libdyld.dylib)[6314] <Notice>: mdmd preparing to stop.`
	All auditable events for the [*not selected]* level of audit	No additional information.	`Dec 5 11:34:58 iPhone-6s mdmd(libdyld.dylib)[6307] <Notice>: mdmd starting…`
	All administrative actions	No additional information.	`Dec 5 12:30:48 iPhone-6s dmd[3038] <Notice>: Received request: <DMFInstallProfileRequest: 0x100c207f0>, from client: <CATTaskSession: 0x100c2f620 { state = Connected, session = BCD262D5-C3B1-4E1F-879C-900ADAFC490E, transport = <CATXPCTransport: 0x100c375b0 { state = Connected }> }>`

	Start-up and shutdown of the Rich OS	No additional information.	`Wed Jan 31 06:24:39 2018 iPhone com.apple.xpc.launchd[1] (com.apple.xpc.launchd.domain.system)`
			`<Notice>: System shutdown initiated by: SpringBoard.63`
			`apcie[2:baseband-pcie]::waitForL2Entry timeout waiting for L2`
			`Wed Jan 31 06:24:46 2018 iPhone com.apple.xpc.launchd[1] (com.apple.xpc.launchd.domain.system)`
			`<Notice>: Userspace teardown took: 6720 ms`
			`Wed Jan 31 06:24:46 2018 iPhone com.apple.xpc.launchd[1] (com.apple.xpc.launchd.domain.system)`
			`<Notice>: Will be calling reboot(2) with flags: 0x8`
			`Kext loading now disabled.`
			`Kext unloading now disabled.`

SFR specified in [ST]	Auditable Events	Additional Audit Record Contents	Example of Audit Records
			Kext autounloading now disabled. syncing disks... Killing all processes flushed all txn's! apfs: total mem allocated: 29123031 (27 mb); all done. going home. (numMountedAPFSVolumes 2) kern_close_file_for_direct_io vnode_close(0) done CPU halted
FCS_STG_EXT.1 {MDF}	Import or key destruction	Identity of key. Role and identity of requestor.	Dec 5 13:38:59 iPhone-6s profiled[93] <Notice>: Removing profile \M-b\M^@\M^\com.apple.mdm.osxserver.atsec.local.9b7bc010-bc15-0135-1603-1801a79c5047.pushed\M-b\M^@\M^]... Dec 5 13:39:00 iPhone-6s profiled[93] <Notice>: Committing restrictions.
FCS_STG_EXT.3 {MDF}	Failure to verify integrity of stored key.	Identity of key being verified.	Sep 25 09:19:38 iPhone securityd[96] <Notice>: Authentication is needed for genp,rowid=49 (-25330): Error Domain=NSOSStatusErrorDomain Code=-25330 "(null)" Sep 25 09:19:38 iPhone securityd[96] <Notice>: Authentication is needed a[280]/1#4 LF=0 copy_matching Error Domain=NSOSStatusErrorDomain Code=-25330 "(null)" Sep 25 09:19:38 iPhone securityd[96] <Notice>: ks_encrypt_data (db): failed: Sep 25 09:19:38 iPhone securityd[96] <Notice>: insert failed for item <genp,rowid=null,cdat=2017-09-25 07:19:38 +0000,mdat=2017-09-25 07:19:38 +0000,desc=null,icmt=null,crtr=null,type=null,scrp=null,labl=null,alis=null,invi=null,nega=null,cusi=null,prot=null,acct=,svce=key_3T,gena=null,agrp=B75W8GX8D3.test.a,pdmn=ck,sync=0,tomb=0,sha1=8756FB888E67AFBFA1E64470E1CED3445A636189,vwht=null

SFR specified in [ST]	Auditable Events	Additional Audit Record Contents	Example of Audit Records
			,tkid=null,v_Data=<?>,v_pk=C7575E0532019C93D96E98F48036B2E68D89F0A2,accc=3120300A0C0470726F740C02636B30120C0361636C310B30090C046461636C010101,u_Tomb=null,musr=,UUID=4EF98D12-E6AF-4E50-B71D-533E99C7066F,sysb=null,pcss=null,pcsk=null,pcsi=null,persistref=> with Error Domain=NSOSStatusErrorDomain Code=-25330 "(null)" UserInfo={-25330=(… Sep 25 09:19:38 iPhone securityd[96] <Notice>: Authentication is needed a[280]/1#4 LF=0 add Error Domain=NSOSStatusErrorDomain Code=-25330 "(null)" UserInfo={-25330=(
FCS_TLSC_EXT.1 {MDF} {AGENT}	Failure to establish an EAP-TLS session.	Reason for failure. Non-TOE endpoint connection.	12/11/17 15:34:37.442 <NOTICE>: Attempting Apple80211AssociateAsync 12/11/17 15:34:37.442 <NOTICE>: Attempting to join EAP network: test 12/11/17 15:34:37.442 <NOTICE>: Completed Apple80211AssociateAsync (-3900 - 0xFFFFF0C4) 12/11/17 15:34:37.442 <ERROR>: Failed to join(-3900 - 0xFFFFF0C4): test
	Establishment/ termination of an EAP-TLS session.	Reason for failure. Non-TOE endpoint connection.	12/11/17 15:34:37.442 <NOTICE>: Attempting Apple80211AssociateAsync 12/11/17 15:34:37.442 <NOTICE>: Attempting to join EAP network: test 12/11/17 15:34:37.442 <NOTICE>: Completed Apple80211AssociateAsync (-3900 - 0xFFFFF0C4) 12/11/17 15:34:37.442 <ERROR>: Failed to join(-3900 - 0xFFFFF0C4): test
FDP_DAR_EXT.1 {MDF}	Failure to encrypt/decrypt data.	No additional information.	Sep 25 09:19:38 iPhone securityd[96] <Notice>: Authentication is needed for genp,rowid=49 (-25330): Error Domain=NSOSStatusErrorDomain Code=-25330 "(null)"

SFR specified in [ST]	Auditable Events	Additional Audit Record Contents	Example of Audit Records
FDP_DAR_EXT.2 {MDF}	Failure to encrypt / decrypt data.		Sep 25 09:19:38 iPhone securityd[96] <Notice>: Authentication is needed a[280]/1#4 LF=0 copy_matching Error Domain=NSOSStatusErrorDomain Code=-25330 "(null)" Sep 25 09:19:38 iPhone securityd[96] <Notice>: ks_encrypt_data (db): failed: Sep 25 09:19:38 iPhone securityd[96] <Notice>: insert failed for item <genp,rowid=null,cdat=2017-09-25 07:19:38 +0000,mdat=2017-09-25 07:19:38 +0000,desc=null,icmt=null,crtr=null,type=null,scrp=null,labl=null,alis=null,invi=null,nega=null,cusi=null,prot=null,acct=,svce=key_3T,gena=null,agrp=B75W8GX8D3.test.a,pdmn=ck,sync=0,tomb=0,sha1=8756FB888E67AFBFA1E64470E1CED3445A636189,vwht=null,tkid=null,v_Data=<?>,v_pk=C7575E0532019C93D96E98F48036B2E68D89F0A2,accc=3120300A0C0470726F740C02636B30120C0361636C310B30090C046461636C010101,u_Tomb=null,musr=,UUID=4EF98D12-E6AF-4E50-B71D-533E99C7066F,sysb=null,pcss=null,pcsk=null,pcsi=null,persistref=> with Error Domain=NSOSStatusErrorDomain Code=-25330 "(null)" UserInfo={-25330=(… Sep 25 09:19:38 iPhone securityd[96] <Notice>: Authentication is needed a[280]/1#4 LF=0 add Error Domain=NSOSStatusErrorDomain Code=-25330 "(null)" UserInfo={-25330=(
FDP_STG_EXT.1 {MDF}	Addition or removal of certificate from Trust Anchor Database	Subject name of certificate	Dec 5 13:38:59 iPhone-6s profiled[93] <Notice>: Removing profile \M-b\M^@\M^\com.apple.mdm.osxserver.atsec.local.9b7bc010-bc15-0135-1603-1801a79c5047.pushed\M-b\M^@\M^]... Dec 5 13:39:00 iPhone-6s profiled[93] <Notice>: Committing restrictions.
FIA_X509_EXT.1 {MDF} {VPN} {AGENT}	Failure to validate x.509v3 certificate	Reason for failure of validation	12/11/17 15:34:37.442 <ERROR>: Failed to join(-3900 - 0xFFFFF0C4): test

SFR specified in [ST]	Auditable Events	Additional Audit Record Contents	Example of Audit Records
FPT_TST_EXT.1/WLAN {WLAN}	Initiation of self-test	No additional information.	SEP: SEP: FIPS POST begin SEP: FIPSPOST_L4 fipspost_post:109: PASSED: (2 ms) - fipspost_post_integrity SEP: sks: FIPS POST Succeeded
FPT_TST_EXT.1/WLAN {WLAN}	Failure of self-test	No additional information.	fipspost_post fipspost_post_integrity -POST_FAILURE: 0xFFFFFFFF
FPT_TST_EXT.2(1) {MDF}	Start-up of TOE	No additional information.	booting kernel at 0x87d703844
FPT_TST_EXT.1/WLAN {WLAN}	Execution of this set of TSF self-test.	No additional information.	corecrypto_kext_start called: tracing enabled FIPSPOST_KEXT fipspost_post:109: PASSED: (0 ms) - fipspost_post_integrity FIPSPOST_KEXT fipspost_post:115: PASSED: (0 ms) - fipspost_post_hmac FIPSPOST_KEXT fipspost_post:117: PASSED: (0 ms) - fipspost_post_aes_ecb FIPSPOST_KEXT fipspost_post:118: PASSED: (0 ms) - fipspost_post_aes_cbc FIPSPOST_KEXT fipspost_post:119: PASSED: (0 ms) - fipspost_post_aes_gcm FIPSPOST_KEXT fipspost_post:120: PASSED: (0 ms) - fipspost_post_aes_xts FIPSPOST_KEXT fipspost_post:121: PASSED: (0 ms) - fipspost_post_tdes_cbc FIPSPOST_KEXT fipspost_post:125: PASSED: (39 ms) - fipspost_post_rsa_sig FIPSPOST_KEXT fipspost_post:126: PASSED: (9 ms) - fipspost_post_ecdsa

SFR specified in [ST]	Auditable Events	Additional Audit Record Contents	Example of Audit Records
			FIPSPOST_KEXT fipspost_post:127: PASSED: (2 ms) - fipspost_post_ecdh FIPSPOST_KEXT fipspost_post:128: PASSED: (0 ms) - fipspost_post_drbg_ctr FIPSPOST_KEXT fipspost_post:129: PASSED: (0 ms) - fipspost_post_drbg_hmac FIPSPOST_KEXT fipspost_post:136: all tests PASSED (129 ms)
FTA_WSE_EXT .1 {WLAN}	All attempts to connect to access points.	Identity of access point being connected to as well as success and failures (including reason for failure)	12/11/17 15:34:37.442 <NOTICE>: Attempting Apple80211AssociateAsync 12/11/17 15:34:37.442 <NOTICE>: Attempting to join EAP network: test 12/11/17 15:34:37.442 <NOTICE>: Completed Apple80211AssociateAsync (-3900 - 0xFFFFF0C4) 12/11/17 15:34:37.442 <ERROR>: Failed to join(-3900 - 0xFFFFF0C4): test
FTP_ITC_EXT.1 /WLAN (3) {WLAN}	All attempts to establish a trusted channel. Detection of modification of channel data.	Identification of the non-TOE endpoint of the channel.	12/11/17 14:00:53.955 <NOTICE>: Attempting Apple80211AssociateAsync 12/11/17 14:00:53.960 <NOTICE>: Attempting to join WPA network: testnet 12/11/17 14:00:54.232 <NOTICE>: Completed Apple80211AssociateAsync (0 - 0x0) 12/11/17 14:00:54.232 <NOTICE>: Joined: testnet 12/11/17 14:01:01.679 <NOTICE>: Update network <testnet>, requested by "configd"
Agent related audit records			
FAU_GEN.1(2) {AGENT}	Start-up and shutdown of the MDM Agent	No additional information.	Dec 5 11:34:58 iPhone-6s mdmd(libdyld.dylib)[6307] <Notice>: mdmd starting… Dec 5 11:39:19 iPhone-6s mdmd(libdyld.dylib)[6314] <Notice>: mdmd preparing to stop.

SFR specified in [ST]	Auditable Events	Additional Audit Record Contents	Example of Audit Records
	Change in MDM policy	No additional information.	`Dec 5 12:30:48 iPhone-6s profiled[93] <Notice>: Profile \M-b\M^@\M^\com.apple.mdm.osxserver.atsec.local.9b7bc010-bc15-0135-1603-1801a79c5047.pushed\M-b\M^@\M^] is replacing an existing profile having the same identifier.`
	Any modification commanded by the MDM Server	No additional information.	`Dec 5 12:30:46 iPhone-6s mdmd(ApplePushService)[6385] <Notice>: Received push notification.`
FAU_ALT_EXT.2 {AGENT}	Type of alert.	No additional information.	`Dec 5 11:34:58 iPhone-6s mdmd(libdyld.dylib)[6307] <Notice>: mdmd starting.`
FAU_SEL.1(2) {AGENT}	All modifications to the audit configuration that occur while the audit collection functions are operating.	No additional information.	`Apr 3 15:43:52 testers-iPhone profiled[156] <Notice>: Profile \M-b\M^@\M^\com.apple.config.testers-MacBook-Pro.local.mdm\M-b\M^@\M^] removed.`
			`Dec 5 13:38:59 iPhone-6s mdmd[6459] <Notice>: Attempting to perform Supervised request: RemoveProfile` `Dec 5 13:38:59 iPhone-6s mdmd(MDM)[6459] <Notice>: Handling request type: RemoveProfile`
			`Dec 5 13:38:59 iPhone-6s profiled[93] <Notice>: Removing profile \M-b\M^@\M^\com.apple.mdm.osxserver.atsec.local.9b7bc010-bc15-0135-1603-1801a79c5047.pushed\M-b\M^@\M^]...` `Dec 5 13:39:00 iPhone-6s profiled[93] <Notice>: Committing restrictions.`

FCS_TLSC_EXT.1 {MDF} {AGENT}	Failure to establish a TLS session	Reason for failure.
		Presented identifier and reference identifier.
		Non-TOE endpoint of connection.
			`12/11/17 15:34:37.442 <NOTICE>: Attempting Apple80211AssociateAsync`
			`12/11/17 15:34:37.442 <NOTICE>: Attempting to join EAP network: test`
			`12/11/17 15:34:37.442 <NOTICE>: Completed Apple80211AssociateAsync (-3900 - 0xFFFFF0C4)`
			`12/11/17 15:34:37.442 <ERROR>: Failed to join(-3900 - 0xFFFFF0C4): test`

SFR specified in [ST]	Auditable Events	Additional Audit Record Contents	Example of Audit Records
	Failure to verify presented identifier	Reason for failure. Presented identifier and reference identifier. Non-TOE endpoint of connection.	12/11/17 15:34:37.442 <NOTICE>: Attempting Apple80211AssociateAsync 12/11/17 15:34:37.442 <NOTICE>: Attempting to join EAP network: test 12/11/17 15:34:37.442 <NOTICE>: Completed Apple80211AssociateAsync (-3900 - 0xFFFFF0C4) 12/11/17 15:34:37.442 <ERROR>: Failed to join(-3900 - 0xFFFFF0C4): test
	Establishment/termination of a TLS session.	Reason for failure. Presented identifier and reference identifier. Non-TOE endpoint of connection.	Dec 5 14:22:42 iPhone-6 mdmd(CFNetwork)[6344] <Notice>: TIC TLS Event [1:0x101107b40]: 1, Pending(0)
FIA_ENR_EXT.2 {AGENT}	Enrollment in management	Reference identifier of MDM Server	Dec 5 12:30:48 iPhone-6s profiled[93] <Notice>: Checking for MDM installation... Dec 5 12:30:48 iPhone-6s profiled[93] <Notice>: ...finished checking for MDM installation.
FMT_POL_EXT.2 {AGENT}	Failure of policy validation.	Reason for failure of validation.	Dec 11 15:33:45 iPhone-5 wifid[41] <Notice>: WiFi:[534720825.769043]: Failed to join(-3906 - 0xFFFFF0BE): test Dec 11 15:33:45 iPhone-5 wifid[41] <Notice>: WiFi:[534720825.780136]: Failed to associate with test, reason -3906
FMT_SMF_EXT.3 {AGENT}	Success or failure of function.	No additional information.	Dec 5 12:30:46 iPhone-6s mdmd(ApplePushService)[6385] <Notice>: Received push notification.

SFR specified in [ST]	Auditable Events	Additional Audit Record Contents	Example of Audit Records

FTP_ITC_EXT.1 (2) {AGENT} {MDF}	Initiation and termination of trusted channel.	Trusted channel protocol. Non-TOE endpoint of connection.	`12/11/17 14:00:53.955 <NOTICE>: Attempting Apple80211AssociateAsync`
			`12/11/17 14:00:53.960 <NOTICE>: Attempting to join WPA network: testnet`
			`12/11/17 14:00:54.232 <NOTICE>: Completed Apple80211AssociateAsync (0 - 0x0)`
			`12/11/17 14:00:54.232 <NOTICE>: Joined: testnet`

Table 15: Audit Record Format

4.2 Audit Storage

Audit records cannot be directly accessed by device users, administrators or MDM administrators on the iOS device regardless of the device's configuration. [AConfig] describes how to use the mobile device console to see all logged. The device console is a function within Apple Configurator 2. While viewing the log files, Administrators have capabilities such as: marking selections, clearing the window to view specific events, or saving the log for troubleshooting.

Additionally, audit records cannot be modified in any way. All audit records can be synced to an MDM application using a Configuration Profile or manually via a trusted workstation using the Apple Configurator 2.

Depending on the underlying OS of the trusted workstation or MDM server, all of the mobile device audit records are transferred to the following locations.

macOS

- ~/Library/Logs/CrashReporter/MobileDevice/[Your_Device_Name]/

Windows:

- C:\Users\[Your_User_Name]\AppData\Roaming\AppleComputer\Logs\CrashReporter\MobileDevice\[Your_Device_Name]\

Audit records are not confined by a global capacity limit and are instead predefined individual services depending on what information is being captured. More information may be found in [IOS_LOGS].

iOS has a logging framework that is used to configure different logging levels for the various iOS subsystems. This framework is configured by creating and installing a logging configuration profile property list file (i.e., .plist file) into the appropriate directory. More information may be found in [LOGGING].

There is no configuration required for audit log locations. since audit logs are stored in the locations specified in this section, by default. These locations cannot be changed.

If unified logging is used, log messages are written to centralized data store on disk instead of in different directories as text log files. More information may be found in [LOGGING].

4.3 Configure the Auditable Items

According to [IOS_LOGS], additional logs can be specified by performing user actions on a device or through using a Configuration Profile. The table below shows which audit logs can be optionally gathered, and how they can be initiated.

Log type	Device user	Configuration Profile
3rd Party Apps for iOS	Instructions	
Accounts/AuthKit for iOS	Instructions	Profile
Ad Platforms for iOS	Instructions	Profile
AirTraffic for iOS	Instructions	

Log type	Device user	Configuration Profile
APNS (Apple Push Notification Service) for iOS	Instructions	Profile
App Store/iTunes Store for iOS	Instructions	Profile
Apple Pay for iOS	Instructions	Profile
Baseband for iOS	Instructions	Profile
Battery Life for iOS	Instructions	Profile
Bluetooth for iOS	Instructions	Profile
Calendar/Reminders for iOS	Instructions	Profile
Carousel for iOS	Instructions	
CarPlay for iOS	Instructions	Profile
Charles Logs for iOS	Instructions	
CloudKit for iOS	Instructions	Profile
Console Logs for iOS	Instructions	
Contacts Data Export for iOS	Instructions	
Continuity (IDS) for iOS	Instructions	Profile
CoreMedia (HTTP Live Streaming) for iOS	Instructions	Profile
Crash Logs for iOS	Instructions	
Device-specific Information for iOS	Instructions	
Disk Space Diagnostics (FSMetadata) for iOS	Instructions	Profile
Enterprise SSO and Kerberos for iOS	Instructions	Profile
FaceTime for iOS	Instructions	Profile
Handoff for iOS	Instructions	
HangTracer (Slow UI)	Instructions	Profile
Health Database Extraction for iOS	Instructions	
HealthKit for iOS	Instructions	Profile
Home app/HomeKit for iOS	Instructions	Profile
iAP for iOS	Instructions	Profile
iCloud Backup for iOS	Instructions	Profile
iCloud Drive for iOS	Instructions	Profile
iCloud Key Value for iOS	Instructions	Profile
iCloud Photo ~~Library~~ for iOS	Instructions	Profile
iWork for iOS	Instructions	Profile
Location Services for iOS	Instructions	Profile
Mail for iOS	Instructions	Profile
Mail Raw Source for iOS	Instructions	

Log type	Device user	Configuration Profile
Mail Sync Diagnostics for iOS	Instructions	
Managed Configuration (MDM) for iOS	Instructions	Profile
Maps for iOS	Instructions	Profile
Media Player for iOS	Instructions	
Messages for iOS	Instructions	Profile
Multipeer Connectivity for iOS	Instructions	
Music for iOS	Instructions	
Phone (General) for iOS	Instructions	Profile
Photos Logging for iOS	Instructions	Profile
Podcasts for iOS	Instructions	
Schoolwork/ClassKit	Instructions	Profile
Screenshots and Screen Recordings for iOS	Instructions	
Siri for iOS	Instructions	Profile
Screenshots and Screen Recordings	Instructions	
Software Update for iOS	Instructions	Profile
Spotlight for iOS	Instructions	Profile
Stackshots for iOS	Instructions	
Sync Diagnostics (DataAccess) for iOS	Instructions	Profile
sysdiagnose for iOS	Instructions	Profile
Tailspin for iOS	Instructions	Profile
TCP Dump for iOS	Instructions	
Test Cases/Sample Projects for iOS	Instructions	
TestFlight for iOS	Instructions	Profile
Touch ID for iOS	Instructions	Profile
Unlock for iOS	Instructions	
Updater for iOS	Instructions	
VPN (Network Extension) for iOS	Instructions	Profile
Wallet for iOS	Instructions	Profile
Wi-Fi for iOS	Instructions	Profile

Table 16: Additional Audit Logs

5 Installed Apps

Table 17: Built-in and Preinstalled Apps lists the Built-in and pre-installed applications on the mobile devices. Those marked “Built in” cannot be removed. Those marked “Preinstalled” are included with purchased devices but may be removed by the user or administrator.

Devices purchased in accordance with section 2.2.1 Obtaining the mobile device(s) do not include any other third party applications when purchased.

App Name	iPad	iPhone
Camera	Built In	Built In
Photos	Built In	Built In
Health		Built In
Messages	Built In	Built In
Phone		Built In
FaceTime	Built In	Built In
Mail	Built In	Built In
Music	Built In	Built In
Wallet		Built In
Safari	Built In	Built In
Maps	Built In	Built In
Siri	Built In	Built In
Calendar	Built In	Built In
iTunes Store	Built In	Built In
App Store	Built In	Built In
Notes	Built In	Built In
News	Built In	Built In
Photo Booth	Built In	
Contacts	Built In	Built In
Books	Built In	Built In
Home	Built In	Built In
Weather		Built In
Reminders	Built In	Built In
Clock	Built In	Built In
TV	Built In	Built In
Stocks	Built In	Built In
Calculator		Built In
Voice Memos	Built In	Built In

App Name	iPad	iPhone
Compass		Built In
Podcasts	Built In	Built In
Watch		Built In
Tips		Built In
Find My iPhone	Built In	Built In
Find My friends	Built In	Built In
Settings		Built In
Files	Built In	Built In
Measure	Built In	Built In
Pages	Preinstalled	Preinstalled
Numbers	Preinstalled	Preinstalled
Keynote	Preinstalled	Preinstalled
iMovie	Preinstalled	Preinstalled
GarageBand	Preinstalled	Preinstalled
iTunes U	Preinstalled	Preinstalled
Clips	Preinstalled	Preinstalled
Apple Store app	Preinstalled	Preinstalled

Table 17: Built-in and Preinstalled Apps

6 References

Table 1: Guidance Documents, contains the references to the guidance documents used when configuring the mobile devices. Below are the references documents providing further more detailed technical information.

[BT] Specification of the Bluetooth System
https://www.bluetooth.com/specifications

[PP_MD_V3.1] U.S. Government Approved Protection Profile - Protection Profile for Mobile Device Fundamentals, Version 3.1
https://www.niap-ccevs.org/Profile/Info.cfm?id=417

[EP_MDM_AGENT_V3.0] U.S. Government Approved Protection Profile - Extended Package for Mobile Device Management Agents Version 3.0
https://www.niap-ccevs.org/Profile/Info.cfm?id=403

[PP_WLAN_CLI_EP_V1.0] Extended Package for WLAN Client Version 1.0
https://www.niap-ccevs.org/Profile/Info.cfm?id=386

[MOD_VPN_CLI_EP_V2.1] PP-Module for VPN Client Version 2.1
https://niap-ccevs.org/Profile/Info.cfm?PPID=419&id=419

[CORECRYPTO]
Apple CoreCrypto Cryptographic Module for ARM, v9.0: FIPS 140-2 Non-Proprietary Security Policy
https://csrc.nist.gov/projects/cryptographic-module-validation-program/

[CORECRYPTO_KERNEL]
Apple CoreCrypto Kernel Cryptographic Module for ARM, v9.0: FIPS 140-2 Non-Proprietary Security Policy
https://csrc.nist.gov/projects/cryptographic-module-validation-program/

[SECURE_KEY_STORE]
Apple Secure Key Store Cryptographic Module, v9.0: FIPS 140-2 Non-Proprietary Security Policy
https://csrc.nist.gov/projects/cryptographic-module-validation-program/

7 Abbreviations and Acronyms

AES	Advanced Encryption Standard
API	Application Programming Interface
CA	Certificate of Authority
CBC	Cypher Block Chaining
CC	Common Criteria
CCM	Counter with CBC-MAC
CRL	Certificate Revocation List
DAR	Data-at-Rest
DEK	Data Encryption Key
DEP	Device Enrollment Program
DES	Data Encryption Standard
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
EAP	Extensible Authentication Protocol
EAP-TLS	Extensible Authentication Protocol-Transport Layer Security
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EP	Extended Package
GCM	Galois/Counter Mode
GSM	Global System for Mobile Communications
HMAC	Keyed-Hash Message Authentication Code
IKE	Internet Key Exchange
IV	Initialization Vector
JSON	JavaScript Object Notation
JTAG	Joint Test Action Group
KEK	Key Encryption Key
L2TP	Layer Two Tunneling Protocol
MDF	Mobile Device Fundamentals
MDM	Mobile Device Management
NITZ	Network Identity and Time Zone
NFC	Near Field Communication
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OTA	Over-the-Air
PAE	Port Access Entity
PBKDF	Password Based Key Derivation Function
PKCS	Public Key Cryptography Standards
PKI	Public Key Infrastructure
PP	Protection Profile
REK	Root Encryption Key
RSA	Rivest-Shamir-Adleman
SA	Secure Association
SCEP	Simple Certificate Enrollment Protocol
SEP	Secure Enclave Processor
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SPD	Security Policy Database
SSL	Secure Sockets Layer
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
UUID	Universally Unique Identifier
VPN	Virtual Private Network
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protected Access
XML	Extensible Markup Language