Apple Inc.

Apple iOS 10.2 VPN Client Security Target

July 2017
Version 1.0

VID: 10792

Prepared for:
Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
www.apple.com

Prepared by:
Acumen Security, LLC.
18504 Office Park Drive
Montgomery Village, MD 20886
www.acumensecurity.net

Table of Contents

Revision History

Version	Date	Description
1.0	July 2017	Initial Release

1 Security Target Introduction

1.1 Security Target and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

Category	Identifier
ST Title	Apple iOS VPN Client Security Target
ST Version	1.0
ST Date	July 2017
ST Author	Acumen Security, LLC.
TOE Identifier	Apple IOS 10.2 VPN Client on iPhone and iPad Note: The TOE is the VPN Client software only. The Apple iOS operating system has been separately validated (VID 10782).
TOE Software Version	10.2
TOE Developer	Apple Inc.
Key Words	VPN, IPsec, Mobility

Table 1 TOE/ST Identification

1.2 TOE Overview

The TOE is the Apple iOS VPN Client which runs on iPad and iPhone devices. The IPsec VPN allows users the ability to have confidentiality, integrity, and protection of data in transit regardless of the transport mechanism (cellular or WiFi).

Note: The TOE is the VPN Client software only. The Apple iOS operating system has been separately validated (VID 10782).

1.3 TOE Description

The TOE is a VPN client on a mobile operating system. The TOE is the VPN Client software only. The Apple iOS operating system has been separately validated (VID 10782). The mobile operating system and hardware platforms are part of the TOE environment. When deployed, the TOE provides a tunnel to a VPN Gateway. The evaluated version of the TOE is version 10.2.

As evaluated, the TOE software runs on the following devices,

Device Name	Model	Processor	WiFi	Cellular	Bluetooth
iPhone 5s	A1533 (GSM) A1533 (CDMA) A1453 A1457 A1530	A7	802.11/a/b/g/n/ac 802.11/a/b/g/n/ac 802.11/a/b/g/n/ac 802.11/a/b/g/n/ac 802.11/a/b/g/n/ac	See next table	4.0 4.0 4.0 4.0 4.0
iPhone 6 Plus/ iPhone 6	A1549/A1522 (GSM) A1549/A1522 (CDMA) A1586/A1524	A8	802.11/a/b/g/n/ac 802.11/a/b/g/n/ac 802.11/a/b/g/n/ac	See next table	4.0 4.0 4.0
iPhone 6S Plus/ iPhone 6S	A1634/A1633 (US) A1687/A1688 (Global)	A9	802.11/a/b/g/n/ac 802.11/a/b/g/n/ac	See next table	4.2 4.2

Device Name	Model	Processor	WiFi	Cellular	Bluetooth
iPhone 7 Plus/ iPhone 7	A1784/A1778 (GSM) A1661/A1660 (CDMA)	A10	802.11/a/b/g/n/ac 802.11/a/b/g/n/ac	See next table	4.2 4.2
iPhone SE	A1662 (US) A1723 (Global)	A9	802.11/a/b/g/n/ac	See next table	4.2
iPad mini 3	A1599 (WiFi only) A1600 (WiFi + cellular) A1601 (WiFi + cellular)	A7	802.11a/b/g/n 802.11a/b/g/n 802.11a/b/g/n	See next table	4.0 4.0 4.0
iPad mini 4	A1538 (WiFi only) A1550 (WiFi + cellular)	A8	802.11a/b/g/n 802.11a/b/g/n	See next table	4.2 4.2
iPad Air 2	A1566 (WiFi only) A1567 (WiFi + cellular)	A8X	802.11a/b/g/n/ac 802.11a/b/g/n/ac	See next table	4.2 4.2
iPad Pro 12.9”	A1584 (WiFi only) A1652 (WiFi + cellular)	A9X	802.11a/b/g/n/ac 802.11a/b/g/n/ac	See next table	4.2 4.2
iPad Pro 9.7”	A1673 (WiFi only) A1674 (WiFi + cellular)	A9X	802.11a/b/g/n/ac 802.11a/b/g/n/ac	See next table	4.2 4.2

Table 2 Devices Covered by the Evaluation

Device Name	Model	Cellular
iPhone 5s	A1533 (GSM)	UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz); LTE (Bands 1, 2, 3, 4, 5, 8, 13, 17, 19, 20, 25)
	A1533 (CDMA)	CDMA EV-DO Rev. A and Rev. B (800, 1700/2100, 1900, 2100 MHz); UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz); LTE (Bands 1, 2, 3, 4, 5, 8, 13, 17, 19, 20, 25)
	A1453	CDMA EV-DO Rev. A and Rev. B (800, 1700/2100, 1900, 2100 MHz); UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz); LTE (Bands 1, 2, 3, 4, 5, 8, 13, 17, 18, 19, 20, 25, 26)
	A1457	UMTS/HSPA+/DC-HSDPA (850, 900, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz); LTE (Bands 1, 2, 3, 5, 7, 8, 20)
	A1530	UMTS/HSPA+/DC-HSDPA (850, 900, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz); FDD-LTE (Bands 1, 2, 3, 5, 7, 8, 20); TD-LTE (Bands 38, 39, 40)
iPhone 6 Plus/ iPhone 6	A1549/A1522 (GSM)	UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz) GSM/EDGE (850, 900, 1800, 1900 MHz) LTE (Bands 1, 2, 3, 4, 5, 7, 8, 13, 17, 18, 19, 20, 25, 26, 28, 29)
	A1549/A1522 (CDMA)	CDMA EV-DO Rev. A and Rev. B (800, 1700/2100, 1900, 2100 MHz) UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz) GSM/EDGE (850, 900, 1800, 1900 MHz) LTE (Bands 1, 2, 3, 4, 5, 7, 8, 13, 17, 18, 19, 20, 25, 26, 28, 29)
	A1586/A1524	CDMA EV-DO Rev. A and Rev. B (800, 1700/2100, 1900, 2100 MHz)

Device Name	Model	Cellular
		UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz) TD-SCDMA 1900 (F), 2000 (A) GSM/EDGE (850, 900, 1800, 1900 MHz) FDD-LTE (Bands 1, 2, 3, 4, 5, 7, 8, 13, 17, 18, 19, 20, 25, 26, 28, 29) TD-LTE (Bands 38, 39, 40, 41)
iPhone 6S Plus/ iPhone 6S	A1634/A1633 (US)	CDMA EV-DO Rev. A (800, 1700/2100, 1900, 2100 MHz) UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz) TD-SCDMA 1900 (F), 2000 (A) GSM/EDGE (850, 900, 1800, 1900 MHz) LTE (Bands 1, 2, 3, 4, 5, 7, 8, 12, 13, 17, 18, 19, 20, 25, 26, 27, 28, 29, 30) TD-LTE (Bands 38, 39, 40, 41)
	A1687/A1688 (Global)	CDMA EV-DO Rev. A (800, 1700/2100, 1900, 2100 MHz) UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz) TD-SCDMA 1900 (F), 2000 (A) GSM/EDGE (850, 900, 1800, 1900 MHz) LTE (Bands 1, 2, 3, 4, 5, 7, 8, 12, 13, 17, 18, 19, 20, 25, 26, 27, 28, 29) TD-LTE (Bands 38, 39, 40, 41)
iPhone 7 Plus/ iPhone 7	A1784/A1778 (GSM)	UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz) GSM/EDGE (850, 900, 1800, 1900 MHz) FDD-LTE (Bands 1, 2, 3, 4, 5, 7, 8, 12, 13, 17, 18, 19, 20, 25, 26, 27, 28, 29, 30) TD-LTE (Bands 38, 39, 40, 41)
	A1661/A1660 (CDMA)	CDMA EV-DO Rev. A (800, 1900, 2100 MHz) UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz) TD-SCDMA 1900 (F), 2000 (A) GSM/EDGE (850, 900, 1800, 1900 MHz) FDD-LTE (Bands 1, 2, 3, 4, 5, 7, 8, 12, 13, 17, 18, 19, 20, 25, 26, 27, 28, 29, 30) TD-LTE (Bands 38, 39, 40, 41)
iPhone SE	A1662 (US)	CDMA EV-DO Rev. A (800, 1700/2100, 1900, 2100 MHz) UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz) GSM/EDGE (850, 900, 1800, 1900 MHz) LTE (Bands 1, 2, 3, 4, 5, 8, 12, 13, 17, 18, 19, 20, 25, 26, 29)
	A1723 (Global)	CDMA EV-DO Rev. A (800, 1700/2100, 1900, 2100 MHz) UMTS/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz) TD-SCDMA 1900 (F), 2000 (A) GSM/EDGE (850, 900, 1800, 1900 MHz) LTE (Bands 1, 2, 3, 4, 5, 7, 8, 12, 17, 18, 19, 20, 25, 26, 28) TD-LTE (Bands 38, 39, 40, 41)
iPad mini 3	A1600	UMTS/HSPA/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz) CDMA EV-DO Rev. A and Rev. B (800, 1900 MHz) LTE (Bands 1, 2, 3, 4, 5, 7, 8, 13, 17, 18, 19, 20, 25, 26)
	A1601	UMTS/HSPA/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz) CDMA EV-DO Rev. A (800, 1900 MHz) TD-SCDMA (1900 (F), 2000 (A)) LTE (Bands 1, 2, 3, 4, 5, 7, 8, 18, 19, 20) TD-LTE (Bands 38, 39, 40)
iPad mini 4	A1550	UMTS/HSPA/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz) CDMA EV-DO Rev. A and Rev. B (800, 1900 MHz) LTE (Bands 1, 2, 3, 4, 5, 7, 8, 13, 17, 18, 19, 20, 25, 26, 28, 29, 38, 39, 40, 41)
iPad Air 2	A1567	GSM/EDGE (850, 900, 1800, 1900 MHz),

Device Name	Model	Cellular
		UMTS/HSPA/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz), CDMA EV-DO Rev. A and Rev. B (800, 1900 MHz), TD-SCDMA LTE (Bands 1, 2, 3, 4, 5, 7, 8, 13, 17,18, 19, 20, 25, 26, 28, 29) TD-LTE (Bands 38, 39, 40,41)
iPad Pro 12.9”	A1652	CDMA EV-DO Rev. A and Rev. B (800, 1900 MHz) UMTS/HSPA/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz) LTE (Bands 1, 2, 3, 4, 5, 7, 8, 13, 17, 18, 19, 20, 25, 26, 28, 29, 38, 39, 40, 41)
iPad Pro 9.7”	A1674	CDMA EV-DO Rev. A and Rev. B (800, 1900 MHz) UMTS/HSPA/HSPA+/DC-HSDPA (850, 900, 1700/2100, 1900, 2100 MHz); GSM/EDGE (850, 900, 1800, 1900 MHz) LTE Advanced (Bands 1, 2, 3, 4, 5, 7, 8, 12, 13, 17, 18, 19, 20, 25, 26, 27, 28, 29, 30, 38, 39, 40, 41)

Table 3 Cellular Protocols Supported

The Operating System on which the TOE is running is Apple iOS version 10. This is the same version of iOS which has undergone Common Criteria evaluation against the Protection Profile for Mobile Device Fundamentals Version 3.0. There are two cryptographic modules on the platforms on which the TOE runs, Apple iOS CoreCrypto Kernel Module v7.0 and Apple iOS CoreCrypto module v7.0. These provide all TOE required cryptographic services.

1.4 TOE Architecture

1.4.1 Physical Boundaries

The TOE is a software application running on a mobile device (as listed above). The mobile device platform provides a host Operating System, controls that limit application behavior, and wireless connectivity. Note: The TOE is the VPN Client software only. The Apple iOS operating system has been separately validated (VID 10782).

Network connectivity for the TOE is provided by the host OS and connects to either 802.11-2012 access points or mobile data networks.

1.4.2 Security Functions provided by the TOE

The TOE provides the security functionality required by [VPNPP].

1.4.2.1 Cryptographic Support

The TOE provides IPsec VPN functionality for clients wishing to securely communicate with remote parties over unsecured networks. The TOE supports IPsec sessions established using IKEv2. The VPN tunnels are configured and controlled by Network Extension Framework, which is a part of the host operating system’s Core OS Layer.

There are two cryptographic modules on the platforms on which the TOE runs, Apple iOS CoreCrypto Kernel Module v7.0 and Apple iOS CoreCrypto Module v7.0. Both cryptographic modules are used together on each platform/processor. These provide all TOE required cryptographic services including, cryptographic key generation, key storage, AES encryption and decryption, cryptographic signature services, cryptographic hashing, keyed-hash message authentication and random bit generation. User Data Protection

The TOE zeroizes all memory used to store packet contents upon reallocation for another purpose.

1.4.2.2 Identification and Authentication
All validation of X.509 certificates is performed by the iOS platform that the TOE is running on.

1.4.2.3 Security Management
The TOE provides the ability to manage all security functionality required by the Protection Profile via the use of configuration files (profiles).

1.4.2.4 Protection of the TSF
The TOE platform performs cryptographic self-tests at startup which ensures the TOE ability to properly operate. The TOE platform also verifies all software updates via digital signature.

1.4.2.5 Trusted Path/Channels
The TOE is an IPsec VPN client. The TOE has the ability to establish IKEv2/IPsec protected communications with VPN gateways.

1.5 Operational Environment
In its evaluated configuration, the TOE is designed to support users in an enterprise setting by providing always-on connectivity via IPSec VPN tunnel in order to provide secure, reliable access to enterprise assets while on the go. To that end, certain elements of IT Infrastructure are utilized

1.5.1 IT Infrastructure
The following elements of IT Infrastructure are assumed to be present:

- A VPN Gateway to service connections from the TOE
- Mobile Device Management (MDM) system
    - In order to operate in the evaluated configuration, the device must be "supervised" and enrolled in some MDM platform, capable of configuring and publishing the necessary Configuration Profile payload to the device
- A PKI system
    - If the TOE will be utilizing x509 certificates for authenticating to the VPN connection, then an enterprise PKI system will need to be in place with the following features:
        i. A CA trusted by both the VPN gateway and the TOE Platform
        ii. An OCSP responder or published CRL to service revocation checking requests.

1.6 TOE Documentation
[VPNPP] Protection Profile for IPsec Virtual Private Network (VPN) Clients, Version 1.4

2 Conformance Claims
2.1 CC Conformance
This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 4, September 2012: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 4, September 2012: Part 3

2.2 Protection Profile Conformance

This TOE is conformant to:

- Protection Profile for IPsec Virtual Private Network (VPN) Clients, Version 1.4, 23 October 2013 [VPNPP].

2.3 Technical Decisions

The following technical decisions were considered as part of this evaluation:

- TD0140: FCS_IPSEC_EXT.1.12, Test 1 - Importing of Private Key and Certificate
- TD0138: IPsec VPN Client Testing of SPD Rules
- TD0124: Auditable Events in VPN IPSEC Client PP
- TD0107: FCS_CKM - ANSI X9.31-1998, Section 4.1.for Cryptographic Key Generation
- TD0097: VPN Gateway selection for FCS_IPSEC_EXT.1.14
- TD0079: RBG Cryptographic Transitions per NIST SP 800-131A Revision 1
- TD0053: Removal of FCS_IPSEC_EXT.1.12 Test 5 from VPN IPSEC Client v1.4
- TD0042: Removal of Low-level Crypto Failure Audit from PPs
- TD0037: IPSec Requirement_DN Verification
- TD0014: Satisfying FCS_IPSEC_EXT.1.13 in VPN GW EP

2.4 Conformance Rationale

This Security Target provides exact conformance to Version 1.4 of the IPsec Virtual Private Network (VPN) Clients Protection Profile. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile performing only operations defined there.

3 Security Problem Definition

The security problem definition has been taken from [VPNPP] and is reproduced here for the convenience of the reader.

3.1 Unauthorized Access to User and TOE Data (T.UNAUTHORIZED_ACCESS)

[VPNPP] does not include requirements that can protect against an insider threat. Authorized users are not considered hostile or malicious and are trusted to follow appropriate guidance. Only authorized personnel should have access to the client device. Therefore, the primary threat agents are the unauthorized entities that try to gain access to the protected network.

The gateway endpoint of the network communication can be both geographically and logically distant from the TOE, and pass through a variety of other systems. These intermediate systems may be under the control of the adversary, and offer an opportunity for communications over the network to be compromised.

Plaintext communication over the network may allow critical data (such as passwords, configuration settings, and user data) to be read and/or manipulated directly by intermediate systems, leading to a compromise of the TOE. IPsec can be used to provide protection for this communication; however, there are a myriad of options that can be implemented for the protocol to be compliant to the protocol specification listed in the RFC. Some of these options can have negative impacts on the security of the connection. For instance, using a weak encryption algorithm (even one that is allowed by the RFC, such as DES) can allow an adversary to read and even manipulate the data on the encrypted channel, thus circumventing countermeasures in place to prevent such attacks. Further, if the protocol is implemented with little-used or non-standard options, it may be compliant with the protocol specification but will not be able to interact with other, diverse equipment that is typically found in large enterprises.

Even though the communication path is protected, there is a possibility that the remote VPN Gateway could be duped into thinking that a malicious third-party user or system is the TOE. For instance, a middleman could intercept a connection request to the TOE, and respond to the VPN Gateway as if it were the TOE. In a similar manner, the TOE could also be duped into thinking that it is establishing communications with a legitimate remote VPN Gateway when in fact it is not. An attacker could also mount a malicious man-in-the-middle-type of attack, in which an intermediate system is compromised, and the traffic is proxied, examined, and modified by this system. This attack can even be mounted via encrypted communication channels if appropriate countermeasures are not applied. These attacks are, in part, enabled by a malicious attacker capturing network traffic (for instance, an authentication session) and “playing back” that traffic in order to fool an endpoint into thinking it was communicating with a legitimate remote entity.

3.2 Inability to Configure the TSF (T.TSF_CONFIGURATION)

Configuring VPN tunnels is a complex and time-consuming process, and prone to errors if the interface for doing so is not well-specified or well-behaved. The inability to configure certain aspects of the interface may also lead to the mis-specification of the desired communications policy or use of cryptography that may be desired or required for a particular users’ site. This may result in unintended weak or plain-text communications while the user thinks that their data are being protected. Other aspects of configuring the TOE or using its security mechanisms (for example, the update process) may also result in a reduction in the trustworthiness of the VPN client.

3.3 Malicious Updates (T.UNAUTHORIZED_UPDATE)

Since the most common attack vector used involves attacking unpatched versions of software

containing well-known flaws, updating the VPN client is necessary to ensure that changes to threat environment are addressed. Timely application of patches ensures that the client is a “hard target”, thus increasing the likelihood that product will be able to maintain and enforce its security policy. However, the updates to be applied to the product must be trustable in some manner; otherwise, an attacker can write their own “update” that instead contains malicious code of their choosing, such as a rootkit, bot, or other malware. Once this “update” is installed, the attacker then has control of the system and all of its data.

Methods of countering this threat typically involve hashes of the updates, and potentially cryptographic operations (e.g., digital signatures) on those hashes as well. However, the validity of these methods introduces additional threats. For instance, a weak hash function could result in the attacker being able to modify the legitimate update in such a way that the hash remained unchanged. For cryptographic signature schemes, there are dependencies on

1) the strength of the cryptographic algorithm used to provide the signature, and
2) the ability of the end user to verify the signature (which typically involves checking a hierarchy of digital signatures back to a root of trust (a certificate authority)).

If a cryptographic signature scheme is weak, then it may be compromised by an attacker and the end user will install a malicious update, thinking that it is legitimate. Similarly, if the root of trust can be compromised, then a strong digital signature algorithm will not stop the malicious update from being installed (the attacker will just create their own signature on the update using the compromised root of trust, and the malicious update will then be installed without detection).

3.4 User Data Disclosure (T.USER_DATA_REUSE)

Data traversing the TOE could inadvertently be sent to a different user; since these data may be sensitive, this may cause a compromise that is unacceptable. The specific threat that must be addressed concerns user data that is retained by the TOE in the course of processing network traffic that could be inadvertently re-used in sending network traffic to a user other than that intended by the sender of the original network traffic.

3.5 TSF Failure (T. TSF_FAILURE)

Security mechanisms of the TOE generally build up from a primitive set of mechanisms (e.g., memory management, privileged modes of process execution) to more complex sets of mechanisms. Failure of the primitive mechanisms could lead to a compromise in more complex mechanisms, resulting in a compromise of the TSF.

4 Security Objectives

The security objectives have been taken from [VPNPP] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the TOE

4.1.1 Establish VPN Tunnels

To address the issues concerning transmitting sensitive data between the TOE and the VPN Gateway described in Section 2.1, compliant TOEs will provide an encrypted channel for these communication paths between themselves and the VPN Gateway. These channels are implemented using IPsec. IPsec is specified by RFCs that offer a variety of implementation choices. Requirements have been imposed on some of these choices (particularly those for cryptographic primitives) to provide interoperability and resistance to cryptographic attack. While compliant TOEs must support all of the choices specified in the ST, they may support additional algorithms and protocols. If such additional mechanisms are not evaluated, guidance must be given to the administrator to make clear the fact that they are not evaluated.

In addition to providing protection from disclosure (and detection of modification) for the communications, IPsec offers two-way authentication of each endpoint in a cryptographically secure manner, meaning that even if there was a malicious attacker between the two endpoints, any attempt to represent themselves to either endpoint of the communications path as the other communicating party would be detected. This authentication is done using X.509 certificates to provide greater assurance in the authentication than is the case with pre-shared keys (although pre-shared keys may be supported by compliant TOEs as long as the use of certificates is also supported). The requirements on the IPsec protocol, in addition to the structure of the protocol itself, provides protection against replay attacks such as those described in Section 2.1.

(O.VPN_TUNNEL → FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.2, FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_IPSEC_EXT.1, FIA_X509_EXT.1, FTP_ITC.1)

4.1.2 Configuration of the TOE

To address the issues concerning the configuration of the TOE described in Section 2.2, the TOE will provide interfaces to control the configuration of IPsec and the underlying cryptographic mechanisms supporting the protocol, management of X.509 certificates, and updates to the TOE.

(O.TOE_CONFIGURATION → FMT_SMF.1)

4.1.3 Verifiable Updates

As outlined in Section 2.3, failure to verify that updates to the client can be trusted may lead to compromise of the security functionality. A first step in establishing trust in the update is to publish a hash of the update that can be verified prior to installing the update. While this establishes that the update downloaded is the one associated with the published hash, it does not indicate if the source of the update/hash combination has been compromised or can't be trusted. To establish trust in the source of the updates, a cryptographic mechanisms and procedures to procure the update, check the update cryptographically through the TOE-provided digital signature mechanism, and install the update will be provided.

(O.VERIFIABLE_UPDATES → FPT_TUD_EXT.1, FCS_COP.1(2), FCS_COP.1(3), FIA_X509_EXT.1)

4.1.4 Residual Information Clearing

In order to counter the threat that user data is inadvertently included in network traffic not intended by the original sender, the TSF ensures that network packets sent from the TOE do not include data "left over" from the processing of previous network information.

(O.RESIDUAL_INFORMATION_CLEARING → FDP_RIP.2)

4.1.5 TSF Self-Test

In order to detect some number of failures of underlying security mechanisms used by the TSF, the TSF will perform self-tests. The extent of this self-testing is left to the product developer, but a more comprehensive set of self-tests should result in a more trustworthy platform on which to develop enterprise architecture.

(O.TSF_SELF_TEST → FPT_TST_EXT.1)

4.2 Security Objectives for the Operational Environment

4.2.1 OE.NO_TOE_BYPASS

Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE.

4.2.2 OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the operational environment.

4.2.3 OE.TRUSTED_CONFIG

Personnel configuring the TOE and its operational environment will follow the applicable security configuration guidance.

5 Security Requirements

This section identifies the Security Functional Requirements for the TOE and/or Platform. The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, dated: September 2012 and all international interpretations.

5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.

Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. Formatting conventions outside of operations matches the formatting specified within the PP.

5.2 Security Functional Requirements for the VPN Client (TOE)

Security functional requirements in the main body of the PP are divided into those that must be satisfied by the VPN client (the TOE), and those that must be satisfied by either the TOE or the platform on which it runs. This section contains the requirements that must be met by the TOE.

5.2.1 Class: Security Management (FMT)

Specification of Management Functions (FMT_SMF)

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1: The TOE shall be capable of performing the following management functions:

- **Specify VPN gateways to use for connections,**
- **Specify client credentials to be used for connections,**
- **[*No additional management functions*]**

5.3 Security Functional Requirements for the VPN Client or Client Platform

Security functional requirements in the main body of the PP are divided into those that must be satisfied by the VPN client (the TOE), and those that must be satisfied by either the TOE or the platform on which it runs. This section contains requirements that must be met, but they can either be met by the TOE or the platform on which the TOE operates.

5.3.1 Class: Cryptographic Support (FCS)

FCS_CKM.1(1) Cryptographic Key Generation (Asymmetric Keys)

FCS_CKM.1.1(1): **Refinement:** The [TOE platform] shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with

- *NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes*

Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;

- *NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [no other curves] (as defined in FIPS PUB 186-4, "Digital Signature Standard")*
- NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes]

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits. See NIST Special Publication 800-57, "Recommendation for Key Management" for information about equivalent key strengths.

FCS_CKM.1 (2) Cryptographic Key Generation (for asymmetric keys - IKE)

FCS_CKM.1.1(2) **Refinement**: The [TOE platform] shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with a: [

- FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes;
- FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and [no other curves]];

and specified cryptographic key sizes *equivalent to, or greater than, a symmetric key strength of 112 bits*.

FCS_CKM_EXT.2 Cryptographic Key Storage

FCS_CKM_EXT.2.1: The [TOE platform] shall store persistent secrets and private keys when not in use in platform-provided key storage.

FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_CKM_EXT.4.1 **Refinement:** The [TOE] shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

Cryptographic Operation (FCS_COP)

FCS_COP.1(1) Cryptographic Operation (Data Encryption/Decryption)

FCS_COP.1.1(1) **Refinement:** The [TOE platform] shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm *AES operating in* **GCM and CBC mode** with cryptographic key sizes 128-bits and 256-bits that meets the following:

- **FIPS PUB 197, "Advanced Encryption Standard (AES)"**
- **NIST SP 800-38D, NIST SP 800-38A.**

FCS_COP.1(2) Cryptographic Operation (for cryptographic signature)

FCS_COP.1.1(2) **Refinement:** The [TOE platform] shall perform **cryptographic signature services** in accordance with a specified cryptographic algorithm:

- [FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA scheme
- FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and [no other curves]]

and cryptographic key sizes [***equivalent to, or greater than, a symmetric key strength of 112 bits***]**.**

FCS_COP.1(3) Cryptographic Operation (Cryptographic Hashing)

FCS_COP.1.1(3) **Refinement:** The [TOE platform] shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512] **and message digest sizes** [160, 256, 384, 512] **bits** that meet the following: *FIPS Pub 180-4, "Secure Hash Standard."*

FCS_COP.1(4) Cryptographic Operation (Keyed-Hash Message Authentication)

FCS_COP.1.1(4) **Refinement:** The [TOE platform] shall perform **keyed-hash message authentication** in accordance with a specified cryptographic algorithm **HMAC-**[SHA-1, SHA-256, SHA-384, SHA-512], key size **[128 to 256 bit], and message digest size of** [160, 256, 384, 512] **bits** that meet the following: **FIPS PUB 198-1, "The Keyed-Hash Message Authentication Code", and FIPS PUB 180-4, "Secure Hash Standard"**.

Extended: Internet Protocol Security (IPsec) Communications

FCS_IPSEC_EXT.1

FCS_IPSEC_EXT.1.1: The [TOE] shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2: The [TOE] shall implement [**tunnel mode**].

FCS_IPSEC_EXT.1.3: The [TOE] shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.4: The [TOE] shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-GCM-128, AES-GCM-256 as specified in RFC 4106, [AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, no other algorithms].

FCS_IPSEC_EXT.1.5: The [TOE] shall implement the protocol: [IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and [no other RFCs for hash functions]].

FCS_IPSEC_EXT.1.6: The [TOE] shall ensure the encrypted payload in the [IKEv2] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and [AES-GCM-128, AES-GCM-256 as specified in RFC 5282].

FCS_IPSEC_EXT.1.8: The [TOE] shall ensure that [IKEv2 SA lifetimes can be configured by [an Administrator] based on [length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs]].

FCS_IPSEC_EXT.1.9: The [TOE platform] shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least **[224, 256, or 384]** bits.

FCS_IPSEC_EXT.1.10: The [TOE platform] shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^**[112, 128, or 192-bits]**.

FCS_IPSEC_EXT.1.11: The [TOE] shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and [5 (1536-bit MODP), 20 (384-bit Random ECP), *1(768-bit MODP), 2(1024-bit MODP), 15(3072-bit MODP), 16(4096-bit MODP), 17(6144-bit MODP), 18(8192-bit MODP*].

FCS_IPSEC_EXT.1.12: The [TOE] shall ensure that all IKE protocols perform peer authentication using a [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [no other method].

FCS_IPSEC_EXT.1.13: The [TOE] shall not establish an SA if the distinguished name (DN) contained in a certificate does not match the expected DN for the entity attempting to establish a connection.

FCS_IPSEC_EXT.1.14: The [TOE] shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv2 IKE_SA] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv2 CHILD_SA] connection.

Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT)

FCS_RBG_EXT.1 Extended: Cryptographic operation (Random Bit Generation)

FCS_RBG_EXT.1.1: The [TOE platform] shall perform all deterministic random bit generation services in accordance with [NIST Special Publication 800-90A using: [CTR_DRBG (AES)]].

FCS_RBG_EXT.1.2: The deterministic RBG shall be seeded by an entropy source that accumulates entropy from [a platform-based RBG] with a minimum of [128 bits] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

5.3.2 Class: User Data Protection (FDP)

Residual Information Protection (FDP_RIP)

FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1: The [TOE] shall enforce that any previous information content of a resource is made unavailable upon the [allocation of the resource to] all objects.

5.3.3 Class: Identification and Authentication (FIA)

X509 Certificates (FIA_X509_EXT)

FIA_X509_EXT.1 Extended: X.509 Certificate Validation

FIA_X509_EXT.1.1: The [TOE platform] shall validate certificates in accordance with the following rules:

- Perform RFC 5280 certificate validation and certificate path validation.
- Validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759].
- Validate the certificate path by ensuring the basicConstraints extension is present and the CA flag is set to TRUE for all CA certificates.
- Validate the extendedKeyUsage field according to the following rules:
    - Certificates used for [trusted updates, integrity verification] shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3).

FIA_X509_EXT.1.2: The [TOE platform] shall only treat a certificate as a CA certificate if the following

is met: the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 Extended: X.509 Certificate Use and Management

FIA_X509_EXT.2.1: The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec exchanges, and [digital signatures for FPT_TUD_EXT.1, integrity checks for FPT_TST_EXT.1.2].

FIA_X509_EXT.2.2: When a connection to determine the validity of a certificate cannot be established, the [TOE platform] shall [not accept the certificate].

FIA_X509_EXT.2.3: The [TOE platform] shall not establish an SA if a certificate or certificate path is deemed invalid.

5.3.4 Class: Security Management (FMT)

Specification of Management Functions (FMT_SMF)

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1: The [TOE platform] shall be capable of performing the following management functions:

- **Configuration of IKE protocol version(s) used,**
- **Configure IKE authentication techniques used,**
- **Configure the cryptoperiod for the established session keys. The unit of measure for configuring the cryptoperiod shall be no greater than an hour,**
- **Configure certificate revocation check,**
- **Specify the algorithm suites that may be proposed and accepted during the IPsec exchanges,**
- **load X.509v3 certificates used by the security functions in [VPNPP],**
- **ability to update the TOE, and to verify the updates,**
- **ability to configure all security management functions identified in other sections of [VPNPP],**
- [no other actions].

5.3.5 Class: Protection of the TSF (FPT)

Extended: TSF Self Test (FPT_TST_EXT)

FPT_TST_EXT.1 Extended: TSF Self Test

FPT_TST_EXT.1.1: The [TOE platform] shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2: The [TOE platform] shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the [*digital signatures*].

Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1 Extended: Trusted Update

FPT_TUD_EXT.1.1: The [TOE platform] shall provide the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2: The [TOE platform] shall provide the ability to initiate updates to TOE firmware/

software.

FPT_TUD_EXT.1.3: The [TOE platform] shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [no other functions] prior to installing those updates.

5.3.6 Class: Trusted Path/Channels (FTP)

Trusted Channel (FTP/ITC)

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 **Refinement:** The [TOE platform] shall **use IPsec** to provide a **trusted** communication channel between itself and **a VPN Gateway** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data**.

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

FTP_ ITC.1.3 The [TOE]] shall initiate communication via the trusted channel *for all traffic traversing that connection*.

5.4 TOE SFR Dependencies Rationale for SFRs

The Protection Profile for IPsec Virtual Private Network (VPN) Clients contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP has been approved.

5.5 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the Protection Profile for IPsec Virtual Private Network (VPN) Clients which are derived from Common Criteria Version 3.1, Revision 4. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

Table 4 Security Assurance Requirements

5.6 Rationale for Security Assurance Requirements

The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated

it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

5.7 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Apple to satisfy the assurance requirements. The table below lists the details.

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked, how potential changes are incorporated, and the degree to which automation is used to reduce the scope for error.
ALC_CMS.1	
ATE_IND.1	Apple will provide the TOE for testing.
AVA_VAN.1	Apple will provide the TOE for testing.

Table 5 TOE Security Assurance Measures

6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE. For additional information regarding functionality provided by the TOE platform, the following document should be referenced, Apple iOS 10.2 MDFPPv2 Security Target, version 1.0.

TOE SFR	Rationale
FCS_CKM.1(1)	The TOE platform provides asymmetric key establishment in support of IPsec VPN using NIST SP 800-56A compliant finite field Diffie-Hellman, NIST SP 800-56A compliant Elliptic Curve Diffie-Hellman, and NIST SP 800-56B compliant RSA Key Transport. The TOE utilizes the cryptographic algorithm implementation of the TOE Platform by linking against the platform cryptographic library. In this way, the TOE can invoke the key establishment operations provided by the TOE Platform. These cryptographic algorithms are used for key establishment the VPN peer. The cryptographic functionality used for IPsec connections is invoked by the TOE by linking to the platform cryptographic modules. These modules, include, Apple iOS CoreCrypto Kernel Module v7.0 and Apple iOS CoreCrypto Module v7.0. When the TOE needs cryptographic services in support of an IKE/IPsec connection, the TOE calls the platform module API. CAVP algorithm certificates were obtained for the TOE's key generation implementation. See the TOE platform ST (VID10782) for details regarding the available algorithm validation certificates (CAVP).
FCS_CKM.1(2)	The TOE platform provides cryptographic signature services using RSA Digital Signatures, as specified in FIPS 186-4, and ECDSA using NIST curves P-256 and P-384, as specified in FIPS 186-4, in support of IKE/IPsec session authentication. These cryptographic algorithms are used for asymmetric key generation. The cryptographic functionality used for IPsec connections is invoked by the TOE by linking to the platform cryptographic modules. These modules, include, Apple iOS CoreCrypto Kernel Module v7.0 and Apple iOS CoreCrypto Module v7.0. When the TOE needs cryptographic services in support of an IKE/IPsec connection, the TOE calls the platform module API. CAVP algorithm certificates were obtained for the TOE's digital signature implementation. See the TOE platform ST (VID10782) for details regarding the available algorithm validation certificates (CAVP).
FCS_CKM_EXT.2	The TOE platform stores all persistent secrets and private keys in the platform's key chain. Specifically, the only two types of persistently stored secrets that are required by the TOE include, • User IPsec X.509v3 Certificate Keys • CA IPsec X.509v3 Certificate Public Keys Both key types are used to facilitate IKE/IPsec protected communications with an IPsec Gateway. User certificate keys are the TOEs certificate keys and the certificate keys of the gateway the TOE is connecting with. IPsec CA certificate Public Keys are used as part of IKE/IPsec session establishment. Each persistent secret is store within the platform's key chain.
FCS_CKM_EXT.4	The TOE and TOE platform in combination meet all requirements for destruction of keys and Critical Security Parameters (CSPs). Please refer to Table 8 "Key Zeroization" for more information on the key zeroization.
FCS_COP.1(1)	The TOE Platform provides symmetric encryption and decryption capabilities in support of IPsec VPN using AES in CBC and GCM mode (with key sizes 128 and 256 bits) as described in

TOE SFR	Rationale
	NIST SP 800-38A and NIST SP 800-38D. These cryptographic algorithms are used for bulk encryption with VPN peers. The cryptographic functionality used for IPsec connections is invoked by the TOE by linking to the platform cryptographic modules. These modules, include, Apple iOS CoreCrypto Kernel Module v7.0 and Apple iOS CoreCrypto Module v7.0. When the TOE needs cryptographic services in support of an IKE/IPsec connection, the TOE calls the platform module API. CAVP algorithm certificates were obtained for the TOE's AES implementation. See the TOE platform ST (VID10782) for details regarding the available algorithm validation certificates (CAVP).
FCS_COP.1(2)	The TOE platform provides cryptographic signature services using RSA Digital Signature Algorithm with key size of 2048 and greater as specified in FIPS PUB 186-4, "Digital Signature Standard," Appendix B.3. In addition, the TOE provides cryptographic signature services using ECDSA with NIST curves P-256 and P-384 as specified in FIPS PUB 186-4, "Digital Signature Standard," Appendix B.4. The cryptographic functionality used for IPsec connections is invoked by the TOE by linking to the platform cryptographic modules. These cryptographic algorithms are used for digital signature validation services. These modules, include, Apple iOS CoreCrypto Kernel Module v7.0 and Apple iOS CoreCrypto Module v7.0. When the TOE needs cryptographic services in support of an IKE/IPsec connection, the TOE calls the platform module API. CAVP algorithm certificates were obtained for the TOE's digital signature implementation. See the TOE platform ST (VID10782) for details regarding the available algorithm validation certificates (CAVP).
FCS_COP.1(3)	The TOE platform provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512 as specified in FIPS Pub 180-4 "Secure Hash Standard." These hashes are used in support of digital signatures for IKE/IPsec authentication and HMACs used to verify the integrity of IKE/IPsec traffic. These cryptographic algorithms are used for hashing when establishing VPN sessions with a VPN peer. The cryptographic functionality used for IPsec connections is invoked by the TOE by linking to the platform cryptographic modules. These modules, include, Apple iOS CoreCrypto Kernel Module v7.0 and Apple iOS CoreCrypto Module v7.0. When the TOE needs cryptographic services in support of an IKE/IPsec connection, the TOE calls the platform module API. CAVP algorithm certificates were obtained for the TOE's hashing implementation. See the TOE platform ST (VID10782) for details regarding the available algorithm validation certificates (CAVP).
FCS_COP.1(4)	The TOE platform provides keyed-hashing message authentication services using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512, as specified in FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code," and FIPS 180-4, "Secure Hash Standard." These cryptographic algorithms are used for MACing when establishing VPN sessions with a VPN peer. In the evaluated configuration, the TOE does not support use cases in which the HMAC hash calculation is truncated (e.g., HMAC-SHA1-96). The HMAC implementation uses the following values for calculating a MAC (varying based on the specific IPsec connection):

TOE SFR	Rationale
	• Key length: 128 to 256 bits • Hash function: SHA-1, SHA-256, SHA-384, and SHA-512 • Block size: 160, 256, 384, 512 bits • Output MAC length: 160, 256, 384, 512 bits The cryptographic functionality used for IPsec connections is invoked by the TOE by linking to the platform cryptographic modules. These modules, include, Apple iOS CoreCrypto Kernel Module v7.0 and Apple iOS CoreCrypto Module v7.0. When the TOE needs cryptographic services in support of an IKE/IPsec connection, the TOE calls the platform module API. CAVP algorithm certificates were obtained for the TOE's MACing implementation. See the TOE platform ST (VID10782) for details regarding the available algorithm validation certificates (CAVP).
FCS_IPSEC_EXT.1.1	The TOE implements the IPsec protocol as specified in RFC 4301. Configuration of VPN connection setting, such as, authentication method and algorithm selection, is performed by the IPsec VPN client administrator. The TOE enforces an "always on" configuration meaning that all traffic entering and leaving the TOE platform interfaces is protected via an IPsec VPN connection. The TOE allows a limited number of services to be configured to either not allow (DISCARD) or be sent plaintext (BYPASS). These services include applications that make use of Captive Networking Identifiers, Voice Mail, and AirPrint. All other communications are always sent through the IPsec tunnel (PROTECT within the SPD). In order to set a service to match a PROTECT rule in the SPD, select "Allow traffic via tunnel." "Drop Traffic" will cause that traffic to match a DISCARD rule. "Allow traffic outside tunnel" will create a BYPASS rule for that service.
FCS_IPSEC_EXT.1.2/ FCS_IPSEC_EXT.1.4/ FCS_IPSEC_EXT.1.5/ FCS_IPSEC_EXT.1.6/ FCS_IPSEC_EXT.1.11	The TOE platform supports the following IPsec connection characteristics, • IKEv2 (as defined in RFCs 5996 and 4307), • Tunnel Mode • Symmetric algorithms for IKE and ESP encryption (AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES-CBC-256), • Integrity mechanisms (HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512), • Key Exchange (Diffie-Hellman Groups 1(768-bit MODP), 2(xx-bit MODP), 5(1536-bit MODP), 14(2048-bit MODP), 15(3072-bit MODP), 16(4096-bit MODP), 17(6144-bit MODP), 18(8192-bit MODP), 19(256-bit Random ECP), and 20 (384-bit Random ECP) Each of these cryptographic mechanisms are provided by one of the following two cryptographic modules, Apple iOS CoreCrypto Kernel Module or Apple iOS CoreCrypto Module.
FCS_IPSEC_EXT.1.3	All data (other than what was described above) is sent through the encrypted tunnel. Any other plaintext data that is received is ignored. This happens automatically without the need to configure an explicit discard.
FCS_IPSEC_EXT.1.7	Not Applicable – The TOE does not support IKEv1. The TOE only supports IKEv2 in the evaluated configuration.
FCS_IPSEC_EXT.1.8	The TOE supports configurable time-based lifetimes for both IKEv2 Phase 1 and Phase 2 SAs. Phase 1 SAs are configurable to 24 hours and phase 2 SAs are configurable to 8 hours. Configuration settings are applied to the TOE via .xml profiles. These profiles can be

TOE SFR	Rationale
	generated via an MDM, an iOS specific tool such as “Apple Configurator,” or by manually editing the .xml file directly.
FCS_IPSEC_EXT.1.9/ FCS_IPSEC_EXT.1.10	The TOE generates the secret value ‘x’ and nonces used in the IKEv2 Diffie-Hellman key exchanges using the TOE platform CAVP validated DRBG specified (as specified in FCS_RBG_EXT.1). The possible lengths of ‘x’ and the nonces are 224, 256, or 384 bits.
FCS_IPSEC_EXT.1.11	The TOE supports the following key exchange groups, Diffie-Hellman Groups 1(768-bit MODP), 2(xx-bit MODP), 5(1536-bit MODP), 14(2048-bit MODP), 15(3072-bit MODP), 16(4096-bit MODP), 17(6144-bit MODP), 18(8192-bit MODP), 19(256-bit Random ECP), and 20 (384-bit Random ECP). The TOE will only negotiate Diffie-Hellman groups based on the configuration applied by the administrator. Groups not included in the configuration profile are not be used.
FCS_IPSEC_EXT.1.12	The supported peer authentication mechanisms, include, RSA or ECDSA X.509v3 digital certificate authentication.
FCS_IPSEC_EXT.1.13	As part of the peer authentication process, a comparison is made of the distinguished name (DN) contained within the peer certificate (specifically the Subject CN, as well as any Subject Alternative Name fields) to the DN of the requested server. If the DN in the certificate does not match the expected DN for the peer, then the session will not be established.
FCS_IPSEC_EXT.1.14	The strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv2/IKE_SA connection and the strength of the symmetric algorithm negotiated to protect the IKEv2 CHILD_SA connection is configured using .xml configuration files. The administrator must explicitly choose the cryptographic algorithms (including key strength) used for each SA. During negotiation the TOE will only negotiate the configured algorithms which must include an IKEv2/IKE_SA at least that of IKEv2 CHILD_SA.
FCS_RBG_EXT.1	The TOE platforms implements a NIST-approved AES-CTR Deterministic Random Bit Generator (DRBG), as specified in SP 800-90. The random number generation used for IPsec connections is invoked by the TOE by linking to the platform cryptographic modules. These cryptographic algorithms are used for random bit generations part of establishing VPN sessions with a VPN peer. These modules, include, Apple iOS CoreCrypto Kernel Module v7.0 and Apple iOS CoreCrypto Module v7.0. When the TOE needs random numbers in support of an IKE/IPsec connection, the TOE calls the platform module API. CAVP algorithm certificates were obtained for the TOE’s random bit generation implementation. See the TOE platform ST (VID10782) for details regarding the available algorithm validation certificates (CAVP). The information on the entropy source has been provided to NIAP as part of this evaluation.
FDP_RIP.2	When the TOE allocates a new packet buffer, the new packet data is used to overwrite any previous data in the buffer. The TOE keeps track of the new packet data size compared to the size of the allocated buffer. If the totality of the new packet data is less than the allocated buffer, the additional allocated space will be filled with zeros prior to sending the packet to its destination.
FIA_X509_EXT.1	The TOE leverages X.509 certificate validation services provided by the TOE platform to validate certificates presented by its VPN peers.

TOE SFR	Rationale
	The X.509 certificates are validated using the certificate path validation algorithm defined in RFC 5280, which can be summarized as follows: • the public key algorithm and parameters are checked • the current date/time is checked against the validity period • revocation status is checked • issuer name of X matches the subject name of X+1 • name constraints are checked • policy OIDs are checked • policy constraints are checked; issuers are ensured to have CA signing bits • path length is checked • critical extensions are processed In order to verify the revocation status of the presented certificates both Certificate revocation Lists (CRLs) and Online Certificate Status Protocol (OCSP) is used.
FIA_X509_EXT.2	X.509v3 certificates are supported for authentication for IPsec VPN connections, code signing for software updates, and software integrity checks. The certificates used for both code signing for software updates, and software integrity checks come pre-installed in ROM during manufacturing. These are the only certificates used for this purpose. The certificate used for IPsec VPN connection are loaded as all other configuration information is loaded, via an .xml configuration file. The TOE will only use the pre-installed certificates for code signing for software updates and software integrity checks. The TOE will only use the configured certificates for IPsec VPN connections. The TOE receives its peer X.509 certificate during the initial establishment of an IPsec tunnel. If during the revocation check of this certificate, the OCSP server cannot contacted, the connection will be established. If the certificate is deemed to be invalid via a revocation check, the communication will cease immediately and a connection will not be established.
FMT_SMF.1	The following security management functions are provided directly by the TOE, • Selection of the VPN gateway, • Presentation of credentials (X.509v3 certificate) used to connect to the gateway. The client itself uses X.509 digital certificates for authenticating to a VPN gateway when establishing an IPsec connection. This certificate may be imported by a user (if allowed by the policy) or installed using configuration profiles. The list of supported certificate and formats, include, • X.509 certificates, • File extensions cer, .crt, .der, .p12, and .pfx The following management functions are provided by the TOE platform. TOE configuration happens by an administrative user using .xml profiles on the TOE platform. These profiles can be created in various ways, including, via an MDM, an iOS specific tool such as “Apple Configurator,” or by manually editing the .xml file directly. The following management functionality can be configured by loading the .xml profiles, • IKE configuration, including, authentication parameters, algorithms, SA lifetimes. • IPsec connection configuration, including, algorithm suites that may be used and SA lifetimes. • Configuration of certificate revocation checking. • Loading of X.509v3 certificates used for IPsec VPN connections.

TOE SFR	Rationale
	The TOE platform directly provides the following management capabilities. • Updating the TOE software, • Verifying TOE software updates
FPT_TST_EXT.1	A suite of power-on self-tests are run during start-up for all cryptographic functionality. These tests include, Software Integrity Test: This test verifies the integrity of the installed software using the 2048-bit RSA signature verified using the key of the installed software integrity certificate. Algorithm Tests for the following algorithms: • AES KAT: These test takes a known plaintext value and encrypts it using a known key. The test is also performed in reverse using a known encrypted value and a known key. • SHS KAT: This test takes a known value hashes that value and compares it to a known hash. • HMAC KAT: This test takes a known value and known key and calculates a MAC. This MAC is compared to a known MAC. • RSA KAT: This test takes a known key pair performs an encrypt and decrypt operation on a known value. The final value is compared to the original value. • ECDSA PWCT: This test takes a key pair and signs a known value. The signature is then verified against the original value. • DRBG KAT: This test takes known seed values and feeds them into the DRBG implementation. The output is compared to the known output value. The implemented self-tests verify both that the software itself hasn't been tamper with or corrupted (ensuring that the functions operate as expected) and that the cryptography (which is vital to the operation of a VPN client) is operating correctly. Both of these tests ensure the product is operating correctly. If all tests successfully complete, the TOE moves to an operational state. In the event that a test fails, the TOE platform kernel will panic, rendering the device inoperable.
FPT_TUD_EXT.1	The TOE platform cryptographically verifies the integrity of all software updates it receives prior to execution. The TOE platform verifies the software update using a PKCS#1 (using SHA-256) signature of the software executable code to ensure that it has not been modified or corrupted. If the integrity test on the software update fails, the TOE will not load the software update. The TOE may only load software originating from Apple and signed with Apple's private signing key. As with all other certificates on the device, the certificate used to sign the software updates are stored encrypted by the platform in the key chain.
FTP_ITC.1	The TOE connects with a remote IPsec VPN Gateway using the IPsec functionality described in FCS_IPSEC_EXT.1 of this table. All communications between the TOE and the IPsec VPN Gateway are protected via this connection.

Table 6 TOE Summary Specification SFR Description

6.1 Key Management

The following table describes the key type, usage, storage, and zeroization referenced for each of the key used by the TOE.

Keys	Type	Usage	Storage	Zeroization Description
DH Group Parameters	Diffie-Hellman	Used as part of IKE/IPsec	RAM	Overwritten with zeros

Keys	Type	Usage	Storage	Zeroization Description
		key establishment		by client
User IPsec X.509v3 Certificate Keys	RSA/EDSA Public/ Private Key Pair	Used to authenticate IKE/IPsec sessions.	Persistently stored encrypted in the platform key chain	Overwritten with zeros by platform
CA IPsec X.509v3 Certificate Public Keys	RSA/EDSA Public Key	Used to authenticate IKE/IPsec sessions.	Persistently stored encrypted in the platform key chain	Overwritten with zeros by platform
IKEv2 IKE_SA Encryption Keys	AES-CBC, AES-GCM	Used to encrypt IKE/IPsec traffic.	RAM	Overwritten with zeros by client
IKEv2 IKE_SA Integrity Keys	HMAC	Used to verify the integrity of IKE/IPsec traffic.	RAM	Overwritten with zeros by client
IKEv2 CHILD_SA Encryption Keys	AES-CBC, AES-GCM	Used to encrypt IKE/IPsec traffic.	RAM	Overwritten with zeros by client
IKEv2 CHILD_SA Integrity Keys	HMAC	Used to verify the integrity of IKE/IPsec traffic.	RAM	Overwritten with zeros by client

Table 7 Key Management