Cisco AnyConnect Secure Mobility Client v4.7 for Android 8

Security Target

Version 1.0

August 23rd 2019

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Table of Contents

List of Tables

List of Figures

List of Acronyms

The following acronyms and abbreviations are common and may be used in this Security Target:

Table 1 Acronyms

Acronyms / Abbreviations	Definition
AES	Advanced Encryption Standard
CC	Common Criteria for Information Technology Security Evaluation
CEM	Common Evaluation Methodology for Information Technology Security
CM	Configuration Management
DRBG	Deterministic Random Bit Generator
EAL	Evaluation Assurance Level
EC-DH	Elliptic Curve-Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ESP	Encapsulating Security Payload
GCM	Galois Counter Mode
HMAC	Hash Message Authentication Code
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
IT	Information Technology
NGE	Next Generation Encryption
OS	Operating System
PP	Protection Profile
PRF	Pseudo-Random Functions
RFC	Request For Comment
SHS	Secure Hash Standard
SPD	Security Policy Database
ST	Security Target
TCP	Transport Control Protocol
TIMA	TrustZone Integrity Measurement Architecture
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Function
TSP	TOE Security Policy
UDP	User datagram protocol
VPN	Virtual Private Network

DOCUMENT INTRODUCTION

Prepared By:
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the Cisco AnyConnect Secure Mobility Client v4.7 for Android and iOS (AnyConnect). This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements, and the IT security functions provided by the TOE which meet the set of requirements. Administrators of the TOE will be referred to as administrators, Authorized Administrators, TOE administrators, semi-privileged, privileged administrators, and security administrators in this document.

REVISION HISTORY

Rev	Date	Description
0.1	November 30th 2018	Initial Draft
0.2	December 19th 2018	Update to add TD0378
0.3	January 8th 2019	Address initial comments
0.4	March 15th 2019	Added TDs
0.5	May 31st, 2019	Address ECS comments
0.6	June 24th 2019	Post testing updates
0.7	July 8th, 2019	Final Updates
0.8	July 10th, 2019	Final Updates
0.9	August 5th, 2019	Updates from Checkout
1.0	August 23rd 2019	Updates from 2nd Checkout

1 SECURITY TARGET INTRODUCTION

The Security Target contains the following sections:

- Security Target Introduction [Section 1]
- Conformance Claims [Section 2]
- Security Problem Definition [Section 3]
- Security Objectives [Section 4]
- IT Security Requirements [Section 5]
- TOE Summary Specification [Section 6]

The structure and content of this ST comply with the requirements specified in the Common Criteria (CC), Part 1, Annex A, and Part 2.

1.1 ST and TOE Reference

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

Table 2: ST and TOE Identification

Name	Description
ST Title	Cisco AnyConnect Secure Mobility Client v4.7 for Android Security Target
ST Version	1.0
Publication Date	August 23rd 2019
Vendor and ST Author	Cisco Systems, Inc.
TOE Reference	Cisco AnyConnect Secure Mobility Client v4.7 for Android
TOE Software Version	4.7
Keywords	VPN Client

1.2 TOE Overview

The TOE is the Cisco AnyConnect Secure Mobility Client v4.7 for Android (herein after referred to as the VPN client, or the TOE). The TOE enables remote users within an organization to communicate securely as if their devices were directly connected to a private network.

The TOE is a VPN Client software application. A virtual private network (VPN) extends the organization's private network across a shared or public network. A VPN client establishes a IKEv2/IPsec connection to a VPN Gateway which allowing the remote user to securely connect to the organization's private network.

TOE Product Type

The TOE product type is a VPN client. A VPN client provides protection of data in transit across a shared or public network. The TOE implements IPsec which establishes a cryptographic tunnel to protect the transmission of data between IPsec peers. The VPN client is intended to be located outside an organization's private network, protecting data flows between a host and the VPN Gateway.

Use case 3 (Communication) as described in [App] and use case 1 (TOE to VPN Gateway) as described in [VPN Client] apply to the TOE.

1.3 TOE DESCRIPTION

This section provides an overview of the Target of Evaluation (TOE). The Cisco AnyConnect TOE is a client application that provides remote users a secure VPN tunnel to protect data in transit on both IPv4 and IPv6 networks. The TOE provides IPsec to authenticate and encrypt network traffic travelling across an unprotected public network. By protecting the communication from unauthorized disclosure or modification, remote users can securely connect to an organization's network resources and applications.

Required non-TOE Hardware/ Software/ Firmware

The TOE requires the following IT environment components when the TOE is configured in its evaluated configuration:

Table 3: Required IT Environment Components

Component	Usage/Purpose Description
Certificate Authority	The Certification Authority provides the TOE with valid certificates. The CA also provides the TOE with a method to check the certificate revocation status of the VPN Gateway.
Android 8 platform	The Android 8 platform provides an execution platform for the TOE to run. The TOE requires a Common Criteria certified Samsung Galaxy Device on Android 8 platform to run. Samsung Galaxy Device 8 devices have been evaluated for conformance with the US Government PP for Mobile Device Fundamentals Version 3.1 as listed on the NIAP Product Compliant List (PCL).
ASA 5500-X series VPN Gateway	The Cisco ASA 5500-X with software version 9.2.2 or later functions as the head-end VPN Gateway. The Cisco AnyConnect TOE communicates only with the Cisco ASA 5500-X Series Gateway.
ASDM Management Platform	The ASDM 7.7 or later operates from any of the following operating systems: • Windows 7, 8, 10 • Windows Server 2008, 2012, 2012 R2 • Apple OS X 10.4 or later • Ubuntu Linux 14.04 • Debian Linux 7 Note that that ASDM software is installed on the ASA appliance and the management platform is used to connect to the ASA and run the ASDM. The only software installed on the management platform is a Cisco ASDM Launcher.

The following figure provides a visual depiction of a TOE deployment.

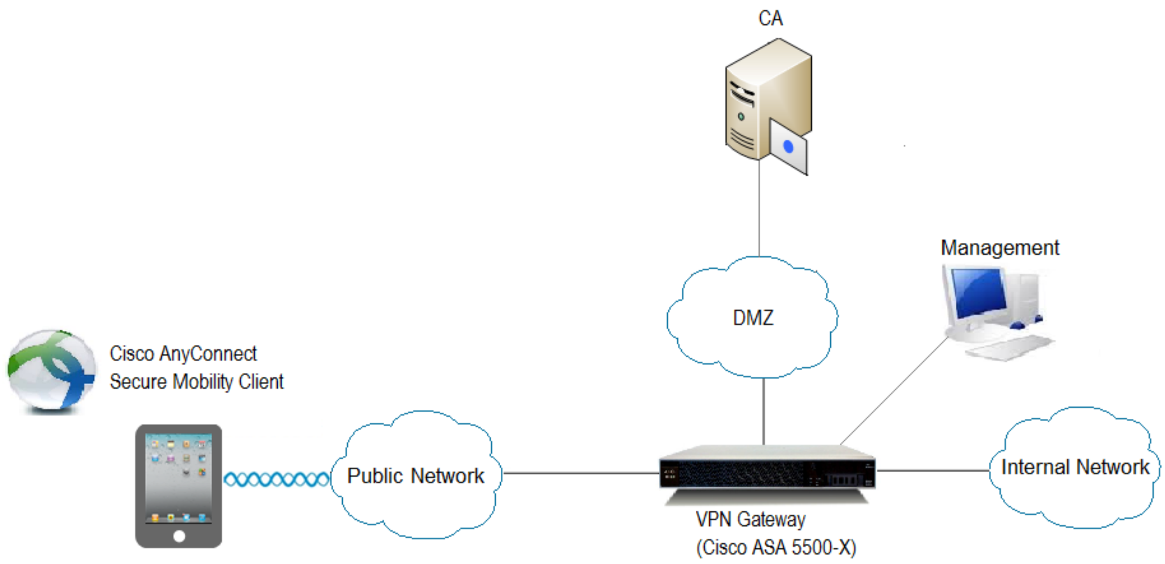

Figure 1 TOE Deployment

1.4 TOE Evaluated Configuration

As a software app, the evaluated configuration is Cisco AnyConnect v4.7 installed on Android 8. Samsung Galaxy Device 8 devices have been evaluated for conformance with the US Government PP for Mobile Device Fundamentals Version 3.1 as listed on the NIAP Product Compliant List (PCL).

Refer to the Common Criteria Administrator's Guide for instructions on installing and configuring the TOE.

1.5 Physical Scope of the TOE

The TOE is a software-only VPN client application. The underlying mobile platform on which the TOE resides is considered part of the IT environment.

1.6 Logical Scope of the TOE

The TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

1. Cryptographic Support
2. User Data Protection
3. Identification and Authentication
4. Security Management
5. Protection of the TSF
6. Trusted Channels

These features are described in more detail in the subsections below.

Cryptographic Support

The TOE incorporates a cryptographic module, CiscoSSL FIPS Object Module, to provide the cryptography in support of IPsec with ESP symmetric cryptography for bulk AES encryption/decryption and SHA-2 algorithm for hashing. In addition the TOE provides the cryptography to support Diffie-Hellman key exchange and the derivation function used in the IKEv2 and ESP protocols. The cryptographic algorithm implementation has been validated for CAVP conformance. See Table 15 in section 7 for certificate references.

The TOE platform provides asymmetric cryptography, which is used by the TOE for IKE peer authentication using digital signature and hashing services. In addition the TOE platform provides a DRBG.

User Data Protection

The TOE platform ensures that residual information from previously sent network packets processed through the platform are protected from being passed into subsequent network packets.

Identification and Authentication

The TOE and TOE platform perform device-level X.509 certificate-based authentication of the VPN Gateway during IKE v2 key exchange. Device-level authentication allows the TOE to establish a secure channel with a trusted VPN Gateway. The secure channel is established only after each endpoint successfully authenticates each other.

Security Management

The TOE, TOE platform, and VPN Gateway provide the management functions to configure the security functionality provided by the TOE.

Protection of the TSF

The TOE performs a suite of self-tests during initial start-up to verify correct operation of its CAVP tested algorithms. Upon execution, the integrity of the TOEs software executables is also verified.

The TOE Platform provides for verification of TOE software updates prior to installation.

Trusted Channels

The TOE's implementation of IPsec provides a trusted channel ensuring sensitive data is protected from unauthorized disclosure or modification when transmitted from the host to a VPN gateway.

1.7 Excluded and Functionality Not Covered

The following functionality is excluded or not covered in the CC evaluation.

Table 4: Excluded and Functionality Not Covered

Functionality	Rationale
Non-FIPS 140-2 mode of operation	This mode of operation includes non-FIPS allowed operations.
SSL Tunnel with DLTS tunneling options	VPNv1.4 Client PP only permits an IPsec VPN tunnel.

The functionality listed above will be disabled by configuration.

2 CONFORMANCE CLAIMS

2.1 Common Criteria Conformance Claim

The TOE and ST are compliant with the Common Criteria (CC) Version 3.1, Revision 5, dated: April 2017. For a listing of Assurance Requirements claimed see section 5.44.

The TOE and ST are CC Part 2 extended and CC Part 3 conformant.

2.2 Protection Profile Conformance

The TOE and ST are conformant with the Protection Profiles as listed in Table 5 below:

Table 5: Protection Profiles

Protection Profile	Version	Date
Protection Profile for Application Software	1.2	April 22, 2016
PP-Module for Virtual Private Network (VPN) Clients	2.1	October 5, 2017

The following table lists NIAP Technical Decisions that are applied to this ST:

NIAP Technical Decision	PP	TOE Applicability	Exclusion Rationale
0435 – Alternative to SELinux for FPT_AEX_EXT.1.3	[App]	No	Linux platform excluded
0434 – Windows Desktop Applications Test	[App]	No	Windows platform excluded
0427 – Reliable Time Source	[App]	Yes	
0392 – FCS_TLSC_EXT.1.2 Wildcard Checking	[App]	No	TLS not claimed
0390 – Cryptographically Secure RNG	[App]	Yes	
0389 – Handling of SSH EP claim for platform	[App]	Yes	
0385 – FTP_DIT_EXT.1 Assurance Activity Clarification	[App]	Yes	
0382 – Configuration Storage Options for Apps	[App]	Yes	
0380 – Linux Keyring Requirement in FCS_STO_EXT.1	[App]	Yes	
0364 – Android mmap testing for FPT_AEX_EXT.1.1	[App]	Yes	
0359 – Buffer Protection	[App]	No	Windows platform excluded
0358 – Cipher Suites for TLS in SWApp v1.2	[App]	No	TLS not claimed
0327 – Default file permissions for FMT_CFG_EXT.1.2	[App]	Yes	
0326 – RSA-based key establishment schemes	[App]	Yes	
0305 – Handling of TLS connections with and without mutual authentication	[App]	No	TLS not claimed
0304 – Update to FCS_TLSC_EXT.1.2	[App]	No	TLS not claimed
0300 – Sensitive Data in FDP_DAR_EXT.1	[App]	Yes	
0296 – Update to FCS_HTTPS_EXT.1.3	[App]	No	HTTPS not claimed
0295 – Update to FPT_AEX_EXT.1.3 Assurance Activities	[App]	Yes	
0268 – FMT_MEC_EXT.1 Clarification	[App]	Yes	
0267 – TLSS testing - Empty Certificate Authorities list	[App]	No	TLS not claimed
0244 – FCS_TLSC_EXT - TLS Client Curves Allowed	[App]	No	TLS not claimed
0241 – Removal of Test 4.1 in FCS_TLSS_EXT.1.1	[App]	No	TLS not claimed
0238 – User-modifiable files FPT_AEX_EXT.1.4	[App]	Yes	

0221 – FMT_SMF.1.1 - Assignments moved to Selections	[App]	No	Applies only when the Extended Package for Software File Encryption is claimed.
0217 – Compliance to RFC5759 and RFC5280 for using CRLs	[App]	Yes	
0215 – Update to FCS_HTTPS_EXT.1.2	[App]	No	HTTPS not claimed
0178 – Integrity for installation tests in AppSW PP	[App]	Yes	
0177 – FCS_TLSS_EXT.1 Application Note Update	[App]	No	TLS not claimed
0174 – Optional Ciphersuites for TLS	[App]	No	TLS not claimed
0172 – Additional APIs added to FCS_RBG_EXT.1.1	[App]	Yes	
0163 – Update to FCS_TLSC_EXT.1.1 Test 5.4 and FCS_TLSS_EXT.1.1 Test	[App]	No	TLS not claimed
0131 – Update to FCS_TLSS_EXT.1.1 Test 4.5	[App]	No	TLS not claimed
0121 – FMT_MEC_EXT.1.1 Configuration Options	[App]	No	Applies only when the Extended Package for Software File Encryption is claimed.
0119 – FCS_STO_EXT.1.1 in PP_APP_v1.2	[App]	Yes	
0107 – FCS_CKM - ANSI X9.31-1998, Section 4.1.for Cryptographic Key Generation	[App]	Yes	
0387 – VPN Client Required SFR for GPOS as Base PP	[VPN Client]	No	ST does not extend the GPOS PP
0385 – FTP_DIT_EXT.1 Assurance Activity Clarification	[VPN Client]	Yes	
0379 – Updated FCS_IPSEC_EXT.1.11 Tests for VPN Client	[VPN Client]	Yes	
0378 – TOE/TOE Platform Selection in FCS_IPSEC_EXT.1 SFRs	[VPN Client]	Yes	
0373 – RSA-based Key Establishment	[VPN Client]	Yes	
0362 – "Failure of the randomization process" audit	[VPN Client]	No	FAU_GEN.1 not claimed
0355 – FCS_CKM.1/VPN for IKE authentication	[VPN Client]	Yes	
0330 – Curve25519 scheme moved to optional and FFC scheme using DH Group 14 added	[VPN Client]	No	Applies only when the MDF PP is the base PP
0303 – IKEv1 and support for XAUTH	[VPN Client]	Yes	

Table 6: NIAP Technical Decisions

2.3 Protection Profile Conformance Claim Rationale

TOE Appropriateness

The TOE provides all of the functionality at a level of security commensurate with that identified in the U.S. Government Protection Profiles:

- Protection Profile for Application Software Version 1.2 [App]
- PP-Module for Virtual Private Network (VPN) Clients Version 2.1 [VPN Client]

TOE Security Problem Definition Consistency

The Assumptions, Threats, and Organization Security Policies included in the Security Target represent the Assumptions, Threats, and Organization Security Policies specified in the Protection Profile for Application Software Version 1.2 and PP-Module for Virtual Private Network (VPN) Clients Version 2.1 for which conformance is claimed verbatim. All concepts covered in the Protection Profile Security Problem Definition are included in the Security Target Statement of Security Objectives Consistency.

The Security Objectives included in the Security Target represent the Security Objectives specified in the Protection Profile for Application Software Version 1.2 and PP-Module for Virtual Private Network (VPN) Clients Version 2.1 for which conformance is claimed verbatim. All concepts covered in the Protection Profile's Statement of Security Objectives are included in the Security Target.

Statement of Security Requirements Consistency

The Security Functional Requirements included in the Security Target represent the Security Functional Requirements specified in the Protection Profile for Application Software Version 1.2 and PP-Module for Virtual Private Network (VPN) Clients Version 2.1 for which conformance is claimed verbatim. All concepts covered the Protection Profile's Statement of Security Requirements are included in the Security Target. Additionally, the Security Assurance Requirements included in the Security Target are identical to the Security Assurance Requirements included in the claimed Protection Profiles.

3 SECURITY PROBLEM DEFINITION

This chapter identifies the following:

- Significant assumptions about the TOE's operational environment.
- IT related threats to the organization countered by the TOE.
- Environmental threats requiring controls to provide sufficient protection.
- Organizational security policies for the TOE as appropriate.

This document identifies assumptions as A.assumption with "assumption" specifying a unique name. Threats are identified as T.threat with "threat" specifying a unique name. Organizational Security Policies (OSPs) are identified as P.osp with "osp" specifying a unique name.

3.1 Assumptions

The specific conditions listed in the following subsections are assumed to exist in the TOE's environment. These assumptions include both practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

Table 7 TOE Assumptions

Assumption	Assumption Definition
A. PLATFORM	The TOE relies upon a trustworthy computing platform with a reliable time clock for its execution. This includes the underlying platform and whatever runtime environment it provides to the TOE.
A.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software in compliance with the applied enterprise security policy.
A.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.
A.NO_TOE_BYPASS	Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_CONFIG	Personnel configuring the TOE and its operational environment will follow the applicable security configuration guidance.

3.2 Threats

The following table lists the threats addressed by the TOE and the IT Environment. The assumed level of expertise of the attacker for all the threats identified below is Enhanced-Basic.

Table 8 Threats

Threat	Threat Definition
T.NETWORK_ATTACK	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may engage in communications with the application software or alter

Threat	Threat Definition
	communications between the application software and other endpoints in order to compromise it.
T.NETWORK_EAVESDROP	An attacker is positioned on a communications channel or elsewhere on the network infrastructure. Attackers may monitor and gain access to data exchanged between the application and other endpoints.
T.LOCAL_ATTACK	An attacker can act through unprivileged software on the same computing platform on which the application executes. Attackers may provide maliciously formatted input to the application in the form of files or other local communications.
T.PHYSICAL_ACCESS	An attacker may try to access sensitive data at rest.
T.UNAUTHORIZED_ACCESS	This PP-Module 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 system or device that contains the IPsec VPN client. Therefore, the primary threat agents are the unauthorized entities that try to gain access to the protected network (in cases where tunnel mode is used) or to plaintext data that traverses the public network (regardless of whether transport mode or tunnel mode is used). The endpoint of the network communication can be both geographically and logically distant from the TOE, and can 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 or to the secured environmental system(s) that the TOE is being used to facilitate communications with. IPsec can be used to provide protection for this communication; however, there are myriad 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 IPsec peer 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 request as if it were the TOE. In a similar manner,

Threat	Threat Definition
	the TOE could also be duped into thinking that it is establishing communications with a legitimate IPsec peer 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.
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 site. This may result in unintended weak or plaintext 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.
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

Threat	Threat Definition
	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).
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.
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.

3.3 Organizational Security Policies

There are no organizational security policies defined in [App] or [VPN Client]

4 SECURITY OBJECTIVES

This Chapter identifies the security objectives of the TOE and the IT Environment. The security objectives identify the responsibilities of the TOE and the TOE's IT environment in meeting the security needs.

- This document identifies objectives of the TOE as O.objective with objective specifying a unique name. Objectives that apply to the IT environment are designated as OE.objective with objective specifying a unique name.

4.1 Security Objectives for the TOE

The following table, Security Objectives for the TOE, identifies the security objectives of the TOE. These security objectives reflect the stated intent to counter identified threats and/or comply with any security policies identified. An explanation of the relationship between the objectives and the threats/policies is provided in the rationale section of this document.

Table 9 Security Objectives for the TOE

Environment Security Objective	TOE Security Objective Definition
O.INTEGRITY	Conformant TOEs ensure the integrity of their installation and update packages, and also leverage execution environment-based mitigations. Software is seldom if ever shipped without errors, and the ability to deploy patches and updates to fielded software with integrity is critical to enterprise network security. Processor manufacturers, compiler developers, execution environment vendors, and operating system vendors have developed execution environment-based mitigations that increase the cost to attackers by adding complexity to the task of compromising systems. Application software can often take advantage of these mechanisms by using APIs provided by the runtime environment or by enabling the mechanism through compiler or linker options.
O.QUALITY	To ensure quality of implementation, conformant TOEs leverage services and APIs provided by the runtime environment rather than implementing their own versions of these services and APIs. This is especially important for cryptographic services and other complex operations such as file and media parsing. Leveraging this platform behavior relies upon using only documented and supported APIs.
O.MANAGEMENT	To facilitate management by users and the enterprise, conformant TOEs provide consistent and supported interfaces for their security-relevant configuration and maintenance. This includes the deployment of applications and application updates through the use of platform-supported deployment mechanisms and formats, as well as providing mechanisms for configuration. This also includes providing control to the user regarding disclosure of any PII.
O.PROTECTED_STORAGE	To address the issue of loss of confidentiality of user data in the event of loss of physical control of the storage medium, conformant TOEs will use data-at-rest protection. This involves encrypting data and keys stored by the TOE in order to prevent unauthorized access to this data. This also includes unnecessary network communications whose consequence may be the loss of data.
O.PROTECTED_COMMS	To address both passive (eavesdropping) and active (packet modification) network attack threats, conformant TOEs will use a trusted channel for sensitive data. Sensitive data includes cryptographic keys, passwords, and any other data specific to the application that should not be exposed outside of the application.

4.2 Security Objectives for the Environment

All of the assumptions stated in section 3.1 are considered to be security objectives for the environment. The following are the Protection Profile non-IT security objectives, which, in addition to those assumptions, are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 10 Security Objectives for the Environment

Environment Security Objective	IT Environment Security Objective Definition
OE.PLATFORM	The TOE relies upon a trustworthy computing platform for its execution. This includes the underlying operating system and any discrete execution environment provided to the TOE.
OE.PROPER_USER	The user of the application software is not willfully negligent or hostile, and uses the software within compliance of the applied enterprise security policy.
OE.PROPER_ADMIN	The administrator of the application software is not careless, willfully negligent or hostile, and administers the software within compliance of the applied enterprise security policy.
OE.NO_TOE_BYPASS	Information cannot flow onto the network to which the VPN client's host is connected without passing through the TOE.
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
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. The Security Functional Requirements in this section are derived from [APP], [VPN_Client] and NIAP Technical Decisions.

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 and ~~strikethroughs~~;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined text*;
- Iteration: Indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

The ST does not identify operations already completed in [App] or [VPN Client].

5.2 TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements that appear in the following table are described in more detail in the following subsections.

Table 11 Security Functional Requirements

Class Name	Component Identification	Component Name	Drawn From
FCS: Cryptographic support	FCS_CKM_EXT.1	Cryptographic Key Generation Services	[VPN Client]
	FCS_CKM.1(1)	Cryptographic Asymmetric Key Generation	[VPN Client]
	FCS_CKM.1/VPN	Cryptographic Asymmetric Key Generation (IKE)	[VPN Client]
	FCS_CKM.2	Cryptographic Key Establishment	[VPN Client]
	FCS_COP.1(1)	Cryptographic Operation – Encryption/Decryption	[App]
	FCS_COP.1(2)	Cryptographic Operation – Hashing	[App]
	FCS_COP.1(3)	Cryptographic Operation – Signing	[App]
	FCS_COP.1(4)	Cryptographic Operation – Keyed–Hash Message Authentication	[App]
	FCS_RBG_EXT.1	Random Bit Generation Services	[App]
	FCS_STO_EXT.1	Storage of Credentials	[App]
	FCS_CKM_EXT.2	Cryptographic Key Storage	[VPN_Client]

Class Name	Component Identification	Component Name	Drawn From
	FCS_IPSEC_EXT.1	IPsec	[VPN_Client]
	FCS_CKM_EXT.4	Cryptographic Key Destruction	[VPN_Client]
FDP: User Data Protection	FDP_DEC_EXT.1	Access to Platform Resources	[App]
	FDP_NET_EXT.1	Network Communications	[App]
	FDP_DAR_EXT.1	Encryption Of Sensitive Application Data	[App]
	FDP_RIP.2	Full Residual Information Protection	[VPN]
FIA: Identification and authentication	FIA_X509_EXT.1	X.509 Certificate Validation	[App]
	FIA_X509_EXT.2	X.509 Certificate Authentication	[App]
FMT: Security management	FMT_MEC_EXT.1	Supported Configuration Mechanism	[App]
	FMT_CFG_EXT.1	Secure by Default Configuration	[App]
	FMT_SMF.1	Specification of Management Functions	[App]
	FMT_SMF.1/VPN	Specification of Management Functions (VPN)	[VPN_Client]
FPR: Privacy	FPR_ANO_EXT.1	User Consent for Transmission of Personally Identifiable Information	[App]
FPT: Protection of the TSF	FPT_API_EXT.1	Use of Supported Services and APIs	[App]
	FPT_AEX_EXT.1	Anti-Exploitation Capabilities	[App]
	FPT_TUD_EXT.1	Integrity for Installation and Update	[App]
	FPT_LIB_EXT.1	Use of Third Party Libraries	[App]
	FPT_TST_EXT.1	TSF Self-Test	[VPN_Client]
FTP: Trusted path/channels	FTP_DIT_EXT.1	Protection of Data in Transit	[VPN_Client]

Class: Cryptographic Support (FCS)

FCS_CKM_EXT.1 Cryptographic Key Generation Services

FCS_CKM_EXT.1.1 The application shall [invoke platform-provided functionality for asymmetric key generation, implement asymmetric key generation].

FCS_CKM.1(1) Cryptographic Asymmetric Key Generation

FCS_CKM.1.1(1) The application shall [implement functionality] to generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- ECC schemes using "NIST curves" P-256, P-384 and [P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;
- [FFC Schemes] using Diffie-Hellman group 14 that meet the following: [RFC 3526, Section 3]];
- [no other key generation methods].

Application Note: *This requirement has applied NIAP TD-0330 and NIAP TD-0373*

FCS_CKM.1/VPN Cryptographic Asymmetric Key Generation (IKE)

FCS_CKM.1.1/VPN The application shall [invoke platform-provided functionality] to generate asymmetric cryptographic used for IKE peer authentication in accordance with: [

- 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 [P-521]]

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

Application Note: *This requirement has applied NIAP TD-0355*

FCS_CKM.2 Cryptographic Key Establishment

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

- Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"; and

- [Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3]; and

- [No other schemes.]

Application Note: *This requirement has applied NIAP TD-0373*

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

FCS_COP.1.1(1) The application shall perform encryption/decryption in accordance with a specified cryptographic algorithm

- AES-CBC (as defined in NIST SP 800-38A) mode; and
- [AES-GCM (as defined in NIST SP 800-38D)]

and cryptographic key sizes 256-bit and [128-bit].

FCS_COP.1(2) – Cryptographic Operation – Hashing

FCS_COP.1.1(2) The application shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm

[SHA-256, SHA-384]

and message digest sizes

[256, 384]

bits that meet the following: FIPS Pub 180-4.

FCS_COP.1(3) – Cryptographic Operation – Signing

FCS_COP.1.1(3) The application shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 4 ,

ECDSA schemes using “NIST curves” P-256, P-384 and [no other curves] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5

].

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

FCS_COP.1.1(4) The application shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm

- HMAC-SHA-256

  and

  [SHA-384]

with key sizes [*256, 384 used in HMAC*] and message digest sizes 256 and [384] bits that meet the following: FIPS Pub 198-1 *The Keyed-Hash Message Authentication Code and FIPS Pub 180-4 Secure Hash Standard.*

FCS_RBG_EXT.1 – Random Bit Generation Services

FCS_RBG_EXT.1.1 The application shall [invoke platform-provided DRBG functionality] for its cryptographic operations.

FCS_STO_EXT.1 – Storage of Credentials

FCS_STO_EXT.1.1 The application shall [invoke the functionality provided by the platform to securely store [*X.509 Certificates*] to non-volatile memory.

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_IPSEC_EXT.1 IPsec

FCS_ IPSEC_EXT.1.1 The [TOE and TOE platform] 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 platform] 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].

FCS_ IPSEC_EXT.1.5 The [TOE] shall implement the protocol: [

- IKEv2 as defined in RFCs 7296 (with mandatory support for NAT traversal as specified in section 2.23), 4307, and [RFC 4868 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.1.7 The [TOE] shall ensure that [IKEv2 SA lifetimes can be configured by [VPN Gateway] based on [length of time]. If length of time is used, it must include at least one option that is 24 hours or less for Phase 1 SAs and 8 hours or less for Phase 2 SAs.

FCS_ IPSEC_EXT.1.8 The [TOE] shall ensure that all IKE protocols implement DH groups 14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP), and [24 (2048-bit MODP with 256-bit POS)].

FCS_ IPSEC_EXT.1.9 The [TOE] 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 [*320 (for DH Group 14), 256 (for DH Group 19), 256 (for DH Group 24), 384 (for DH Group 20)*] bits.

FCS_ IPSEC_EXT.1.10 The [TOE] 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^[*256*].

FCS_ IPSEC_EXT.1.11 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.1.12 The [TOE] shall not establish an SA if the [IP address, Fully Qualified Domain Name (FQDN)] and [no other reference identifier type] contained in a certificate does not match the expected value(s) for the entity attempting to establish a connection.

FCS_ IPSEC_EXT.1.13 The [TOE] shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer.

FCS_ IPSEC_EXT.1.14 The [VPN Gateway] 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.

FCS_CKM_EXT.4 Cryptographic Key Destruction

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

Class: User Data Protection (FDP)

FDP_DEC_EXT.1 Access to Platform Resources

FDP_DEC_EXT.1.1 The application shall restrict its access to [network connectivity].

FDP_DEC_EXT.1.2 The application shall restrict its access to [no sensitive information repositories].

FDP_NET_EXT.1 Network Communications

FDP_NET_EXT.1.1 The application shall restrict network communications to [user-initiated communication for [*IKEv2/IPsec tunnel establishment*]].

FDP_DAR_EXT.1 Encryption Of Sensitive Application Data

FDP_DAR_EXT.1.1 The application shall [protect sensitive data in accordance with FCS_STO_EXT.1] in non-volatile memory.

Application Note: *This requirement has applied NIAP TD-0300*

FDP_RIP.2 Full Residual Information Protection

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

Class: Identification and Authentication (FIA)

FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1.1 The application shall [invoked platform-provided functionality, implement functionality] to validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The application shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The application shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 2560].
- The application shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.
    - Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field.

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

Application Note: *This requirement has applied NIAP TD-0217*

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The application shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and [no other protocols].

FIA_X509_EXT.2.2 When the application cannot establish a connection to determine the validity of a certificate, the application shall [not accept the certificate].

Class: Security Management (FMT)

FMT_MEC_EXT.1 Supported Configuration Mechanism

FMT_MEC_EXT.1.1 The application shall invoke the mechanisms recommended by the platform vendor for storing and setting configuration options.

FMT_CFG_EXT.1 Secure by Default Configuration

FMT_CFG_EXT.1.1 The application shall provide only enough functionality to set new credentials when configured with default credentials or no credentials.

FMT_CFG_EXT.1.2 The application shall be configured by default with file permissions which protect the application's binaries and data files from modification by normal unprivileged user.

Application Note: *This requirement has applied NIAP TD-0327*

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions [no management functions].

FMT_SMF.1 /VPN Specification of Management Functions (VPN)

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

- Specify VPN gateways to use for connections,
- Specify client credentials to be used for connections,
- Configure the reference identifier of the peer

]

Class: Privacy (FPR)

FPR_ANO_EXT.1 – User Consent for Transmission of Personally Identifiable Information

FPR_ANO_EXT.1.1 The application shall [not transmit PII over a network].

Class: Protection of the TSF (FPT)

FPT_API_EXT.1 Use of Supported Services and APIs

FPT_API_EXT.1.1 The application shall use only documented platform APIs.

FPT_AEX_EXT.1 Anti-Exploitation Capabilities

FPT_AEX_EXT.1.1 The application shall not request to map memory at an explicit address except for [no exceptions].

FPT_AEX_EXT.1.2 The application shall [not allocate any memory region with both write and execute permissions].

FPT_AEX_EXT.1.3 The application shall be compatible with security features provided by the platform vendor**.**

FPT_AEX_EXT.1.4 The application shall not write user-modifiable files to directories that contain executable files unless explicitly directed by the user to do so.

FPT_AEX_EXT.1.5 The application shall be compiled with stack-based buffer overflow protection enabled.

FPT_TUD_EXT.1 Integrity for Installation and Update

FPT_TUD_EXT.1.1 The application shall [leverage the platform] to check for updates and patches to the application software.

FPT_TUD_EXT.1.2 The application shall be distributed using the format of the platform-supported package manager.

FPT_TUD_EXT.1.3 The application shall be packaged such that its removal results in the deletion of all traces of the application, with the exception of configuration settings, output files, and audit/log events.

FPT_TUD_EXT.1.4 The application shall not download, modify, replace or update its own binary code.

FPT_TUD_EXT.1.5 The application shall [provide the ability] to query the current version of the application software.

FPT_TUD_EXT.1.6 The application installation package and its updates shall be digitally signed such that its platform can cryptographically verify them prior to installation.

FPT_LIB_EXT.1	Use of Third Party Libraries

FPT_LIB_EXT.1.1 The application shall be packaged with only [

- *OpenSSL*
- *Boost*
- *Knox*
- *gson*
- *libxml*
- *libcurl*

]

FPT_TST_EXT.1	TSF Self-Test

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

FPT_TST_EXT.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 [*cryptographic signature verification service provided by the TOE Platform*].

Class: Trusted Path/Channels (FTP)

FTP_DIT_EXT.1	Protection of Data in Transit

FTP_DIT_EXT.1.1 The application shall encrypt all transmitted sensitive data with **IPsec** and [no other protocols] between itself and another trusted IT product.

5.3 TOE SFR Dependencies Rationale

The [APP] and [VPN Client] contain all the requirements claimed in this Security Target. As such the dependencies are not applicable since the PPs themselves have been approved.

5.4 Security Assurance Requirements

SAR Requirements

The TOE assurance requirements for this ST are taken directly from [APP] and [VPN Client] which are derived from [CC_PART3]. The assurance requirements are summarized in the table below.

Table 12: Assurance Measures

Assurance Class	Components Description
Security Target (ASE)	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
Development (ADV)	Basic functional specification (ADV_FSP.1)
Guidance documents (AGD)	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
Life cycle support (ALC)	Labeling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
	Timely Security Updates (ALC_TSU_EXT.1)
Tests (ATE)	Independent testing – sample (ATE_IND.1)
Vulnerability assessment (AVA)	Vulnerability survey (AVA_VAN.1)

Security Assurance Requirements Rationale

The Security Assurance Requirements (SARs) in this Security Target represent the SARs identified in the [APP] and [VPN Client]. As such, the [APP] and [VPN Client] SAR rationale is deemed acceptable since the PPs themselves have been validated.

5.5 Assurance Measures

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

Table 13: Assurance Measures

Component	How requirement will be met
ADV_FSP.1	No additional "functional specification" documentation was provided by Cisco to satisfy the Evaluation Activities specified in the SD.
AGD_OPE.1 AGD_PRE.1	Guidance must include a description of how the IT personnel verifies that the Operational Environment can fulfill its role for the security functionality. The documentation should be in an informal style and readable by the IT personnel. Guidance must be provided for every operational environment that the product supports as claimed in the ST. This guidance includes: • instructions to successfully install the TSF in that environment; and • instructions to manage the security of the TSF as a product and as a component of the larger operational environment; and • instructions to provide a protected administrative capability. Guidance pertaining to particular security functionality must also be provided. Cisco will provide the guidance documents with the ST.
ALC_CMC.1 ALC_CMS.1	Cisco will identify the TOE such that it can be distinguished from other products or versions from the Cisco and can be easily specified when being procured by an end user.
ALC_TSU_EXT.1	Cisco will provide a Security Vulnerability Policy.
ATE_IND.1	Cisco will provide the TOE for testing.
AVA_VAN.1	Cisco will provide the TOE for Vulnerability Analysis.

6 TOE SUMMARY SPECIFICATION

6.1 TOE Security Functional Requirement Measures

This chapter identifies and describes how the Security Functional Requirements identified in section 5 are met by the TOE.

Table 14: How TOE SFRs Measures

TOE SFRs	How the SFR is Met
FCS_CKM_EXT.1 FCS_CKM.1/VPN	The TOE Platform provides a specified key generation algorithm to generate asymmetric cryptographic keys for IKE authentication. The key sizes are: • RSA scheme: 2048 bit • ECC using NIST curve of P-256, P-384, and P-521 The key generation function is invoked by the TOE platform Administrator using a MDM product.
FCS_CKM_EXT.1 FCS_CKM.1(1)	Key generation for asymmetric keys used by IPsec for key establishment is provided by the TOE and is implemented using ECDSA with NIST curve sizes P-256, P-384, and P-521 according to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4 and FFC using Diffie-Hellman group 14 that meets RFC 3526 section 3.
FCS_CKM.2	To support IPsec the TOE implements the following algorithms to perform key establishment: • ECC key establishment schemes that meet SP800-56A. • DH group 14 key establishment scheme that meets standard RFC 3526, section 3. The TOE implements and uses the prime and generator specified in RFC 3526 Section 3 when generating parameters for the key exchange.
FCS_COP.1(1)	The TOE provides symmetric encryption and decryption capabilities using AES as specified in ISO 18033-3 supporting the following modes: o CBC mode as specified in ISO 10116. o GCM mode as specified in ISO 19772. The TOE uses AES in IPsec using the following modes and key sizes: CBC mode with key size of 128 and 256 bits. GCM mode with key sizes of 128 and 256 bits.
FCS_COP.1(2)	The TOE provides cryptographic hashing services in support of HMAC in IKEv2 and IPsec using SHA-256 and SHA-384 as specified in FIPS Pub 180-3 “Secure Hash Standard.”
FCS_COP.1(3)	The TOE provides cryptographic signature services using RSA Digital Signature Algorithm with key size of 2048 and Elliptic Curve Digital Signature Algorithm with a key size of 256, 384, or 521 bits as specified in FIPS PUB 186-4, “Digital Signature Standard.”
FCS_COP.1(4)	The TOE provides keyed-hashing message authentication services using HMAC-SHA-256 (key size – 256 bits, block size 512 bits) and HMAC-SHA-384 (key size – 384 bits, block size 1024 bits).
FCS_RBG_EXT.1	The TOE invokes /dev/random on the platform when needed to generate a cryptographic key. This applies to the following SFRs: FCS_CKM.2 – Cryptographic Key Establishment FCS_IPSEC_EXT.1 – IPsec Protocol
FCS_STO_EXT.1	The Cisco AnyConnect TOE leverages the platform to store X.509v3 certificates used by the TOE for IKE peer authentication. Certificates are stored in the Android KeyStore.

TOE SFRs	How the SFR is Met
FCS_CKM_EXT.2	The TOE platform stores ECDSA and RSA private keys used by the TOE for IKE peer authentication. Private Keys are stored in the Android KeyStore The TOE does not use pre-shared keys for IPsec.
FCS_IPSEC_EXT.1	The TOE's implementation of the IPsec standard (in accordance with RFC 4301) uses the Encapsulating Security Payload (ESP) protocol to provide authentication, encryption and anti-replay services. By default ESP operates in tunnel mode. No configuration is required by the user or administrator for the TOE to operate in tunnel mode. Remote access policies on the ASA VPN Gateway provide an interface for the administrator to create ACL(s), defining network segment(s) requiring IPsec protection. An XML format of the policy on client defines the remote access policy the TOE will use. After successful client authentication to the ASA VPN Gateway, a "Cisco AnyConnect Secure Mobility Client" virtual interface is created and assigned an IP address from the Gateway's VPN address pool. The TOE's virtual interface includes a kernel mode driver digitally signed by Cisco Systems, Inc. The Security Policy Database (SPD) is implemented by the underlying TOE Platform and the TOE interacts with the SPD through insertions of entries to the routing table on the host OS platform. This enforces what traffic is protected with IPsec by the TOE and what traffic isn't. The default behavior of the remote access policy on the VPN Gateway is for the TOE to protect all traffic with IPsec. When all traffic is tunneled, a new default route is added to the host OS platform with a lower metric directing all traffic to be protected with IPsec by the TOE. The TOE uses active SA settings or creates new SAs for initial connections with the ASA VPN Gateway peer. All ESP processing to authenticate, encrypt, and tunnel the traffic is performed by the TOE. If an organization explicitly permits use of split-tunneling, a remote access policy on the ASA VPN Gateway allows the administrator to define IPsec protection for the organization's network(s) but bypass protection for other traffic. When a portion of traffic is tunneled, a route is added to the host OS platform corresponding to the network segment requiring IPsec protection by the TOE. Network(s) not subjected to the remote access policy, but reachable from the platform, such as Internet traffic, travels without being protected with IPsec by the TOE. SPD discard rules are performed exclusively by the TOE platform. The TOE implements IKEv2 and does not support IKEv1. IPsec Internet Key Exchange is the negotiation protocol that lets the TOE and a VPN Gateway agree on how to build an IPsec Security Association (SA). IKE separates negotiation into two phases: phase 1 and phase 2. During IKE Phase 1, the TOE authenticates the remote VPN Gateway using device-level authentication with ECDSA or RSA X.509v3 certificates provided by the TOE platform. The TOE compares its reference identifier to the identifier presented by the VPN Gateway peer. The TOE supports reference identifiers as configured by the Administrator to be either FQDN or IP address and compares it to the Subject Alternative Name (SAN) or the Common Name (CN) fields in the certificate of the peer. The order of comparison is SAN followed by CN. If the TOE successfully matches the reference identifier to the presented identifier, IKE Phase 1 authentication will succeed. Otherwise it will fail if it does not match. Phase 1 creates the first tunnel, which protects later IKE negotiation messages. The key negotiated in phase 1 enables IKE to communicate securely in phase 2.

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Met</th></tr>
<tr><td></td><td>The TOE supports only IKEv2 session establishment. As part of this support, the TOE by default does not support aggressive mode used in IKEv1 exchanges.
The TOE supports Diffie-Hellman Group 14 (2048-bit keys), 19 (256-bit Random ECP), 24 (2048-bit MODP with 256-bit POS), and 20 (384-bit Random ECP) in support of IKE Key Establishment negotiated in phase 1. These keys are generated using the DRBG specified in FCS_RBG_EXT.1 having 256 bits of entropy.
The administrator is instructed in the AGD to select a supported DH group using one of the following corresponding key sizes (in bits): 320 (for DH Group 14), 256 (for DH Group 19), 256 (for DH Group 24), and 384 (for DH Group 20) bits.
For each DH Group, the TOE generates the secret value 'x' used in the IKEv2 Diffie-Hellman key exchange ('x' in g^x mod p) using its DH private key, the IPsec peer's public key and a nonce. When a random number is needed for a nonce, the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^{256}. The nonce is likewise generated using the DRBG specified in FCS_RBG_EXT.1.
During Phase 2, IKE negotiates the IPsec SA and includes:
• The negotiation of mutually acceptable IPsec SA parameters;
• The Pseudo-Random Function (PRF) is used for the construction of keying material for cryptographic algorithms used in the SA.
• The establishment of IPsec Security Associations to protect packet flows using Encapsulating Security Payload (ESP).
The resulting potential strength of the symmetric key will be 128 or 256 bits of security depending on the algorithms negotiated between the two IPsec peers.
The VPN Gateway ensures by default 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.
After IKE phase 2 completes, the IPsec SA is established, providing a secure tunnel to a remote VPN Gateway. The TOE performs IKEv2 payload and bulk IPsec encryption using AES-GCM-128, AES_GCM-256, AES-CBC-128, or AES-CBC-256 algorithms. The VPN Gateway allows the administrator to configure AES-GCM-128, AES_GCM-256, AES-CBC-128, and AES-CBC-256 encryption algorithms.
The TOE supports administratively configured lifetimes for both Phase 1 SAs and Phase 2 SAs. The default time value for Phase 1 SAs is 24 hours. The value for Phase 2 SAs is configurable to 8 hours. Both values are configurable using management functions provided by the VPN Gateway.</td></tr>
<tr><td>FCS_CKM_EXT.4</td><td>The TOE ensures volatile memory areas containing the following keys are zeroized:
<table>
<tr><th>Key, Secret, or CSP</th><th>Purpose</th><th>Zeroization Method</th></tr>
<tr><td>SK_ei</td><td>IKE SA Initiator Encryption Key</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
<tr><td>SK_er</td><td>IKE SA Responder Encryption Key</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Met</th></tr>
<tr><td></td><td>
<table>
<tr><td>SK_ai</td><td>IKE SA Initiator Integrity Key</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
<tr><td>SK_ar</td><td>IKE SA Responder Integrity Key</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
<tr><td>Diffie-Hellman Shared Secret</td><td>IKE v2 SA setup</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
<tr><td>SK_d</td><td>IKEv2 SA key from which child IPsec keys are derived.</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
<tr><td>Initiator encryption and integrity key</td><td>IPsec child SA key that encrypts and authenticates outgoing ESP traffic.</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
<tr><td>Responder encryption and integrity key</td><td>IPsec child SA key that decrypts and authenticates incoming ESP traffic.</td><td>Overwritten with zeros when no longer in use by the IPsec VPN trusted channel.</td></tr>
</table>
<p>The TOE platform zeroizes private keys it manipulates and stores on the TOE platform:</p>
<table>
<tr><th>Key, Secret, or CSP</th><th>Purpose</th><th>Zeroization Method</th></tr>
<tr><td>Asymmetric ECDSA Private Key stored on the mobile device platform</td><td>ECDSA digital signature generation</td><td>Performed exclusively by the TOE Platform.</td></tr>
<tr><td>Asymmetric RSA Private Key stored on the mobile device platform</td><td>RSA digital signature generation</td><td>Performed exclusively by the TOE Platform.</td></tr>
</table>
</td></tr>
<tr><td>FDP_DEC_EXT.1</td><td>The Cisco AnyConnect TOE restricts access to network connectivity resources.</td></tr>
<tr><td>FDP_NET_EXT.1</td><td>The Cisco AnyConnect TOE limits network communication to user initiated communication for IKEv2/IPsec tunnel establishment</td></tr>
<tr><td>FDP_DAR_EXT.1</td><td>Sensitive data in the TOE is defined as the private key used for X.509 certificate generation and peer authentication, which is protected in accordance with FCS_STO.EXT.1</td></tr>
<tr><td>FDP_RIP.2</td><td>The TOE platform transmits packets over WiFi or cellular radio and therefore is responsible for clearing residual information.</td></tr>
<tr><td>FIA_X509_EXT.1</td><td>The Cisco AnyConnect TOE invokes functionality provided by the platform to perform certificate path validation on the certificate chain presented by ASA VPN Gateway. The certificate path validation begins with the identity certificate presented by ASA VPN Gateway and proceeds through intermediate CA certificate(s) up to a trusted root certificate issued by a trusted certificate</td></tr>
</table>

TOE SFRs	How the SFR is Met
	authority (CA). The following steps are performed for each certificate in the path: • The certificate must not be expired. • The certificate must not be revoked. • The issuer name is checked to ensure it matches the subject name of the previous certificate in the chain. • All CA certificates must have the basicConstraints extension present and be of type CA=TRUE. • The extendedKeyUsage field must be valid based on the following rules: o Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field. o Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field. o Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field. o S/MIME certificates presented for email encryption and signature shall have the Email Protection purpose (id-kp 4 with OID 1.3.6.1.5.5.7.3.4) in the extendedKeyUsage field. o OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field. o Server certificates presented for EST shall have the CMC Registration Authority (RA) purpose (id-kp-cmcRA with OID 1.3.6.1.5.5.7.3.28) in the extendedKeyUsage field The Cisco AnyConnect TOE implements revocation checking itself using OCSP. This includes verifying the OCSP response is signed with a cert that has the OCSP signing purpose. These checks ensure certificate validation results in a trusted root certificate. At any point if a certificate cannot be successfully validated, the AGD guidance instructs the administrator to configure the TOE to not allow the user an option for continuing the connection. In all cases, if a certificate or certificate path cannot be validated, the TOE will not establish an IPsec connection to an untrusted ASA VPN Gateway.
FIA_X509_EXT.2	During TOE installation the user imports a new certificate to the certificate store. The user can also select the certificate used by tapping 'Import' and then 'Device Credential Storage'. The Cisco AnyConnect TOE compares the FQDN of the server it is establishing connectivity with, against the Subject Alternate Name-dnsName attributes in the certificate. If AnyConnect determines there is a mismatch, it will not establish the IPsec trusted channel.
FMT_MEC_EXT.1	All IPsec configuration for the Cisco AnyConnect TOE is stored remotely on the Cisco ASA VPN Gateway. As described in guidance the user controls the following settings which must enabled: “FIPS Mode” “Enable CRL Check"

TOE SFRs	How the SFR is Met
	"Strict Certificate Trust"
FMT_CFG_EXT.1	The Cisco AnyConnect TOE is not installed with any preset default credentials. Users can only access files which are associated to the installation that user performed.
FMT_SMF.1	The Cisco AnyConnect TOE does not perform any security management functions from [App].
FMT_SMF.1/VPN	The Cisco AnyConnect TOE is capable of the following security management functions from [VPN Client]: • Specify VPN gateways to use for connections • Specify client credentials to be used for connections • Configuring the reference identifier of the peer In context of the AnyConnect TOE, client credentials are a X.509 certificate which is used to authenticate the ASA VPN Gateway when authenticating an IPsec session.
FPR_ANO_EXT.1	The Cisco AnyConnect TOE does not transmit PII.
FPT_API_EXT.1	The Cisco AnyConnect TOE uses the following Android APIs: android.app.Activity android.app.KeyguardManager android.app.Notification android.app.NotificationManager android.app.PendingIntent android.app.Service android.content.BroadcastReceiver android.content.ComponentName android.content.ContentValues android.content.Context android.content.Intent android.content.IntentFilter android.content.RestrictionsManager android.content.ServiceConnection android.content.SharedPreferences.Editor android.content.SharedPreferences android.content.pm.ApplicationInfo android.content.pm.PackageManager.NameNotFoundException android.content.pm.PackageManager android.content.pm.ResolveInfo android.content.res.Resources android.net.Credentials android.net.LocalSocket android.net.Proxy android.net.Uri android.net.VpnService android.os.Binder android.os.Build android.os.Bundle android.os.Handler android.os.IBinder android.os.IInterface android.os.Message android.os.ParcelFileDescriptor android.os.Parcelable android.os.Process

TOE SFRs	How the SFR is Met
	android.os.RemoteCallbackList android.os.RemoteException android.provider.Settings android.security.KeyChain android.security.keystore.KeyInfo android.security.keystore.KeyPermanentlyInvalidatedException android.security.keystore.KeyProperties android.security.keystore.KeyProtection android.support.v4.app.NotificationCompat android.text.TextUtils android.widget.Toast java.io.BufferedOutputStream java.io.BufferedReader java.io.ByteArrayInputStream java.io.ByteArrayOutputStream java.io.File java.io.FileDescriptor java.io.FileInputStream java.io.FileNotFoundException java.io.FileOutputStream java.io.IOException java.io.InputStream java.io.InputStreamReader java.io.ObjectInputStream java.io.ObjectOutputStream java.io.Serializable java.io.UnsupportedEncodingException java.lang.reflect.Constructor java.lang.reflect.Method java.net.InetAddress java.nio.charset.Charset java.security.InvalidAlgorithmParameterException java.security.InvalidKeyException java.security.Key java.security.KeyFactory java.security.KeyStore java.security.KeyStoreException java.security.NoSuchAlgorithmException java.security.Principal java.security.PrivateKey java.security.Signature java.security.UnrecoverableKeyException java.security.cert.CertPath java.security.cert.CertPathBuilder java.security.cert.CertPathValidator java.security.cert.CertPathValidatorException java.security.cert.CertStore java.security.cert.CertStoreException java.security.cert.Certificate java.security.cert.CertificateEncodingException java.security.cert.CertificateException java.security.cert.CertificateExpiredException java.security.cert.CertificateFactory java.security.cert.CertificateNotYetValidException

TOE SFRs	How the SFR is Met
	java.security.cert.CertificateParsingException java.security.cert.CollectionCertStoreParameters java.security.cert.PKIXBuilderParameters java.security.cert.PKIXCertPathBuilderResult java.security.cert.PKIXParameters java.security.cert.TrustAnchor java.security.cert.X509CertSelector java.security.cert.X509Certificate java.security.spec.InvalidKeySpecException java.util.ArrayList java.util.Arrays java.util.Collection java.util.Collections java.util.HashMap java.util.HashSet java.util.LinkedHashMap java.util.LinkedList java.util.List java.util.Locale java.util.Map.Entry java.util.Map java.util.Objects java.util.Set java.util.TreeMap java.util.concurrent.CopyOnWriteArraySet java.util.concurrent.CountDownLatch java.util.concurrent.TimeUnit java.util.zip.ZipEntry java.util.zip.ZipInputStream javax.crypto.Cipher javax.net.ssl.SSLException javax.net.ssl.TrustManager javax.net.ssl.TrustManagerFactory javax.net.ssl.X509TrustManager org.apache.http.conn.ssl.StrictHostnameVerifier org.apache.http.conn.ssl.X509HostnameVerifier
FPT_AEX_EXT.1	The Cisco AnyConnect TOE enables ASLR and stack protection by fPIE -pie and the -fstack-protector-all flags.
FPT_TUD_EXT.1 ALC_TSU_EXT.1	The TOE has specific versions that can be queried by a user. A TOE update is not a patch applied to the existing TOE, it is a new version of the TOE. When TOE updates are made available by Cisco, an administrator can obtain and install the update. Upon installation of a TOE update, a digital signature verification check will automatically be performed to ensure it has not been modified since distribution. The authorized source for the digitally signed updates is "Cisco Systems, Inc.". All Cisco communications relating to security issues are handled by the Cisco Product Security Incident Response Team (PSIRT). Cisco aims to provide fixes in 30 days but depending on the timing it may be greater than 30 days though not more than 60 days for most security issues. Fixes may be delayed longer for low-risk security issues. Updates are then made available at Cisco Software Central available at: https://software.cisco.com.

TOE SFRs	How the SFR is Met
	Customers can subscribe to the Cisco Notification Service allows users to subscribe and receive important information regarding product updates. Full information is provide in the Cisco Security Vulnerability Policy available at: https://tools.cisco.com/security/center/resources/security_vulnerability_policy.html
FPT_LIB_EXT.1	The Cisco AnyConnect TOE is packaged with the following third-party libraries: OpenSSL Boost Knox gson libxml libcurl
FPT_TST_EXT.1	As a software product incorporating a cryptographic module, the TOE runs a suite of self-tests during start-up to verify its correct operation. These tests include: • AES Known Answer Test – For the encrypt test, a known key is used to encrypt a known plain text value resulting in an encrypted value. This encrypted value is compared to a known encrypted value to ensure that the encrypt operation is working correctly. The decrypt test is just the opposite. In this test a known key is used to decrypt a known encrypted value. The resulting plaintext value is compared to a known plaintext value to ensure that the decrypt operation is working correctly. • RSA Signature Known Answer Test (both signature/verification) – This test takes a known plaintext value and Private/Public key pair and used the public key to encrypt the data. This value is compared to a known encrypted value to verify that encrypt operation is working properly. The encrypted data is then decrypted using the private key. This value is compared to the original plaintext value to ensure the decrypt operation is working properly. • ECDSA Signature Test – This test takes a known plaintext value and Private/Public key pair and used the public key to encrypt the data. This value is compared to a known encrypted value to verify that encrypt operation is working properly. The encrypted data is then decrypted using the private key. This value is compared to the original plaintext value to ensure the decrypt operation is working properly. • HMAC Known Answer Test– For each of the hash values (256 and 384), the HMAC implementation is fed known plaintext data and a known key. These values are used to generate a MAC. This MAC is compared to a known MAC to verify that the HMAC and hash operations are operating correctly. • SHA Known Answer Test – For each of the values (256 and 384), the SHA implementation is fed known data and key. These values are used to generate a hash. This hash is compared to a known value to verify they match and the hash operations are operating correctly. • Software Integrity Test - The Software Integrity Test is run automatically whenever the module is loaded and confirms the image has maintained its integrity. If any self-test fails subsequent invocation of any cryptographic function calls is prevented. If all components of the power-up self-test are successful then the product is in FIPS mode.

TOE SFRs	How the SFR is Met
	Integrity verification is performed each time the AnyConnect app is loaded and it will wait for the integrity verification to complete. Cryptographic services provided by the TOE platform are invoked to verify the digital signature of the TOE's executable files. If the integrity verification fails to successfully complete, the GUI will not load, rendering the app unusable. If the integrity verification is successful, the app GUI will load and operate normally. These tests are sufficient to verify that the TOE software is operating correctly as well as the cryptographic operations are all performing as expected.
FTP_DIT_EXT.1	The Cisco AnyConnect TOE uses IPsec to encrypt transmitted data.

7 Supplemental TOE Summary Specification Information

See table 15 below for CAVP certificates.

Table 15: CAVP Certificates

SFR	Algorithm	CAVP Certificate Number
FCS_CKM.1(1)	ECDSA	C781
FCS_CKM.1/VPN	ECDSA, RSA	ECDSA 1463 (Samsung) RSA 2937 (Samsung)
FCS_CKM.2	CVL-KAS-ECC	C781
FCS_COP.1(1)	CBC, GCM	C781
FCS_COP.1(2)	SHS	C781
FCS_COP.1(3)	RSA	C781
FCS_COP.1(3)	ECDSA	C781
FCS_COP.1(4)	HMAC	C781

8 ANNEX A: REFERENCES

The following documentation was used to prepare this ST:

Table 16: References

Identifier	Description
[CC_PART1]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-001
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-002
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-003
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated April 2017, version 3.1, Revision 5, CCMB-2017-04-004
[APP]	Protection Profile for Application Software Version 1.2, April 22nd 2016
[VPN_Client]	PP-Module for VPN Client Version 2.1, October 5th, 2017
[SD]	Supporting Document – PP-Module for Virtual Private Network (VPN) Client, Version 2.1, October 2017