D4 Secure VPN Client for the HTC A9 Secured by Cog Systems (IVPNCPP14) Security Target

Version 0.7
October 31, 2017

Prepared for:

Cog Systems
Level 1, 277 King Street
Newtown NSW 2042
Australia

Prepared By:

www.gossamersec.com

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the VPN client for the D4 Secure Communications Mobile phone provided by Cog Systems. The TOE is being evaluated as an IPsec VPN client.

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

<u>Conventions</u>

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a parenthetical number placed at the end of the component. For example FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title – D4 Secure VPN Client for the HTC A9 Secured by Cog Systems (IVPNCPP14) Security Target

ST Version – Version 0.7

ST Date – October 31, 2017

1.2 TOE Reference

TOE Identification – D4 Secure VPN Client for the HTC A9 Secured by Cog Systems

TOE Developer – Cog Systems

Evaluation Sponsor – Cog Systems

1.3 TOE Overview

The Target of Evaluation (TOE) is the D4 Secure VPN Client that is the HTC A9 Secured by Cog Systems D4 Secure Mobile device's built-in Outer Data-In-Transit (DIT) VPN client.

Product	Carrier	Security Software Version	OS version	HTC Software Version number
HTC-A9	Telstra	1.6	Android v6.0.1	1.57.617.52

1.4 TOE Description

The Target of Evaluation (TOE) is the D4 Secure VPN Client that is the HTC A9 Secured by Cog Systems D4 Secure Mobile device's built-in Outer Data-In-Transit (DIT) VPN client. The Outer DIT VPN runs only on the evaluated HTC A9 Secured by Cog Systems D4 Secure Mobile device.

The D4 Secure is a smartphone based upon an HTC A9 hardware which uses Qualcomm SoCs (Snapdragon 617, MSM8952) and runs custom Cog Systems D4 Secure images. This is a custom built smartphone intended to support military and civil service users. The D4 Secure Mobile Device is the TOE Platform for the Outer DIT VPN client. Since the Outer DIT VPN is built-into the evaluated D4 Secure Mobile device, it is considered to have the same version as the D4 Secure Mobile device.

The TOE provides always on secure remote network connectivity for the D4 Secure and Android 6.0.1 operating system, by providing an IPsec VPN that once configured protects all communication. The Outer DIT VPN client sends all network communication to the connected VPN gateway through an IPsec protected communication channel.

1.4.1 TOE Architecture

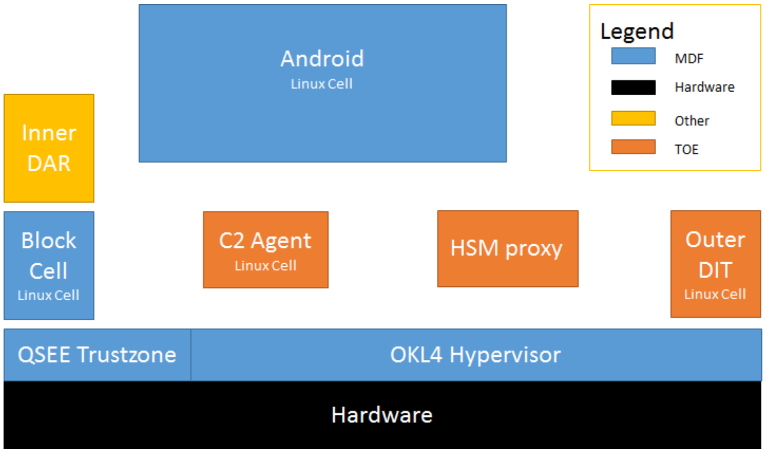
The TOE is a built-in VPN client (referred to as the Outer DIT). The cell providing the TOE's built-in VPN client is referred to as the Outer DIT cell. The TOE also includes a "C2 Agent" cell and an "HSM proxy" cell. These cells cooperate with the Outer DIT cell to facilitate interaction with the TOE Platform[1]. All IPSec protocol functions are provided by the TOE. All network traffic from the Android Cell is passed through the Outer DIT cell by the platform, thus ensuring that the Outer DIT VPN can protect all traffic.

The VPN Client relies upon its platform for random numbers with which it seeds its own DRBG. All cryptography supporting the IPsec protocol stack is provided by the TOE. Data stored by the TOE utilize functions offered by the platform.

The following figure depicts the "Cells" in the D4 Secure Mobile Device. The figure shows the Outer DIT VPN client (i.e., Outer DIT cell, C2 Agent cell and HSM proxy cell), as well as the D4 Secure Mobile cells supporting the Outer DIT VPN Client. The D4 Secure Mobile Device is packaged to include all of the pieces shown in Figure 1-1. The "blue", "yellow" and "black" boxes in Figure 1-1 represent software that is part of the TOE Platform. The TOE is composed of only the cells shown in orange. The Outer DIT cell and C2 agent cell are running a Linux kernel that provides an environment for the cell's functionality. The HSM proxy cell is a cell running customized C language code.

[1] Refer to the Platform Security Target for a description of platform cells.

Figure 1-1 D4 Secure Mobile Architecture



The TOE platform ensures that all network traffic from the Android cell passes through the Outer DIT cell which encapsulates the traffic in an IPsec tunnel. The outer DIT cell is a Linux 3.10.84 kernel with StrongSwan version 5.5.1 IPsec, OpenSSL 2.0.14 cryptographic library, as well as other non-cryptographic supporting libraries. The Outer DIT cell interacts with other cells (specifically the Android cell) via virtualized Ethernet. This ensures that the communication from the Android cell must pass through the Outer DIT cell (irrespective of whether the phone is connected via Wi-Fi or Mobile) and thus through the D4 Secure VPN Client.

The administrator configures the D4 Secure VPN client using a physically connected provisioning workstation. The provisioning workstation directly writes to the phone's internal, non-volatile memory after the user has unlocked the mobile device. The user cannot change the configuration once the device has been provisioned.

1.4.1.1 Physical Boundaries

The D4 Secure Outer DIT VPN Client runs entirely within the outer DIT cell of the D4 Secure mobile device. From a cryptographic perspective, all cryptography is performed using TOE software running in the Outer DIT VPN cell. The Outer DIT VPN cell (the TOE) relies upon the TOE platform for random numbers with which the Outer DIT VPN cell seeds its own DRBG. All subsequent need for random values by Outer DIT VPN cell software obtain those values from the Outer DIT VPN cell’s own DRBG. The Outer DIT VPN cell also relies upon the TOE platform to verify the validity of updates.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by D4 Secure VPN Client:

- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- Trusted path/channels

1.4.1.2.1 Cryptographic support

The IPsec implementation is the primary function of the TOE. IPsec is used by the TOE to protect communication between itself and a VPN Gateway over an unprotected network. The TOE also includes cryptographic services to support the IPsec VPN, and self-testing functionality specified in this Security Target.

1.4.1.2.2 User data protection

The TOE ensures that residual information is protected from potential reuse in accessible objects such as network packets.

1.4.1.2.3 Identification and authentication

The TOE provides the ability to use pre-shared keys and X.509 certificates that are used for IPsec Virtual Private Network (VPN) connections. The TOE utilizes TOE Platform functions to store and protect X.509 certificates.

1.4.1.2.4 Security management

The TOE provides all the interfaces necessary to manage the security functions identified throughout this Security Target to the admin at provisioning. This includes interfaces to the admin as well as to the VPN gateway. The IPsec VPN is fully configurable through a provisioning process performed prior to the first use of the D4 Secure Mobile Device. The TOE platform provides the functions necessary to securely update the TOE.

1.4.1.2.5 Protection of the TSF

The TOE utilizes its own cryptographic functions to perform self-tests that cover the TOE cryptographic operations. The TOE relies upon its underlying platform to perform self-tests that cover the TOE as well as the functions necessary to securely update the TOE.

1.4.1.2.6 Trusted path/channels

The TOE acts as a VPN client using IPsec to established secure channels to corresponding VPN gateways.

1.4.2 TOE Documentation

Cog Systems offers the following documentation to users for the installation and operation of their product. The following list of documents was examined as part of the evaluation.

- D4 Secure VPN Client Guide Documentation, Version 1.1, October 31, 2017.

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
    - Part 3 Conformant
- Package Claims:
    - Protection Profile for IPsec Virtual Private Network (VPN) Clients, Version 1.4, 21 October 2013 (IVPNCPP14)

2.1 Conformance Rationale

The ST conforms to the IVPNCPP14. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

The ST incorporates the following NIAP Technical Decisions:

- TD0107
- TD0037
- TD0079

3. Security Objectives

The Security Problem Definition may be found in the IVPNCPP14 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The IVPNCPP14 offers additional information about the identified security objectives, but that has not been reproduced here and the IVPNCPP14 should be consulted if there is interest in that material.

In general, the IVPNCPP14 has defined Security Objectives appropriate for IPsec VPN client and as such are applicable to the D4 Secure Mobile VPN Client TOE.

3.1 Security Objectives for the Operational Environment

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 operational environment.

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

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the IVPNCPP14. The IVPNCPP14 defines the following extended requirements and since they are not redefined in this ST the IVPNCPP14 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FCS_CKM_EXT.2: Cryptographic Key Storage
- FCS_CKM_EXT.4: Cryptographic Key Zeroization
- FCS_IPSEC_EXT.1: Extended: Internet Protocol Security (IPsec) Communications
- FCS_RBG_EXT.1: Extended: Cryptographic operation (Random Bit Generation)
- FIA_X509_EXT.1: Extended: X.509 Certificate Validation
- FIA_X509_EXT.2: Extended: X.509 Certificate Use and Management
- FPT_TST_EXT.1: Extended: TSF Self Test
- FPT_TUD_EXT.1: Extended: Trusted Update

5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the IVPNCPP14. The refinements and operations already performed in the IVPNCPP14 are not identified (e.g., highlighted) here, rather the requirements have been copied from the IVPNCPP14 and any residual operations have been completed herein. Of particular note, the IVPNCPP14 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the IVPNCPP14 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the IVPNCPP14 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The IVPNCPP14 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by D4 Secure Mobile VPN Client TOE.

Requirement Class	Requirement Component
FCS: Cryptographic support	FCS_CKM.1(1): Cryptographic Key Generation (Asymmetric Keys)
	FCS_CKM.1(2): Cryptographic Key Generation (for asymmetric keys - IKE)
	FCS_CKM_EXT.2: Cryptographic Key Storage
	FCS_CKM_EXT.4: Cryptographic Key Zeroization
	FCS_COP.1(1): Cryptographic Operation (Data Encryption/Decryption)
	FCS_COP.1(2): Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(3): Cryptographic Operation (Cryptographic Hashing)
	FCS_COP.1(4): Cryptographic Operation (Keyed-Hash Message Authentication)
	FCS_IPSEC_EXT.1: Extended: Internet Protocol Security (IPsec) Communications
	FCS_RBG_EXT.1: Extended: Cryptographic operation (Random Bit Generation)
FDP: User data protection	FDP_RIP.2: Full Residual Information Protection
FIA: Identification and authentication	FIA_PSK_EXT.1: Pre-Shared Key Composition
	FIA_X509_EXT.1: Extended: X.509 Certificate Validation
	FIA_X509_EXT.2: Extended: X.509 Certificate Use and Management
FMT: Security management	FMT_SMF.1(1): Specification of Management Functions
	FMT_SMF.1(2): Specification of Management Functions
FPT: Protection of the TSF	FPT_TST_EXT.1: Extended: TSF Self Test
	FPT_TUD_EXT.1: Extended: Trusted Update
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel

Table 5-1 TOE Security Functional Components

5.1.1 Cryptographic support (FCS)

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

FCS_CKM.1(1).1

Refinement: The [***TOE***] 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 [***P-521***] (as defined in FIPS PUB 186-4, 'Digital Signature Standard');
- [***no other***]

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.

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

FCS_CKM.1(2).1

Refinement: The [***TOE***] shall generate asymmetric cryptographic keys used for IKE peer authentication in accordance with a (TD0107 applied): [

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.

5.1.1.3 Cryptographic Key Storage (FCS_CKM_EXT.2)

FCS_CKM_EXT.2.1

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

5.1.1.4 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1

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

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

FCS_COP.1(1).1

Refinement: The [***TOE***] 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.

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

FCS_COP.1(2).1

Refinement: The [***TOE***] 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', Appendix B.4 for ECDSA schemes and implementing 'NIST curves' P-256, P-384 and [P-521]***],

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

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

FCS_COP.1(3).1

Refinement: The [***TOE***] 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.'

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

FCS_COP.1(4).1

Refinement: The [***TOE***] shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC- [***SHA-1, SHA-256, SHA-384, SHA-512***], -key size [***160, 256, 384, 512***], 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'.

5.1.1.9 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 (specified by RFC 3602) together with a Secure Hash Algorithm (SHA)-based HMAC, AES-CBC-256 (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 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.1.7

The [***TOE***] shall ensure that IKEv1 Phase 1 exchanges use only main mode

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***] 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 [**192**] 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^[**96**] .

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 [***24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP)***].

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 [***Pre-Shared Keys***].

FCS_IPSEC_EXT.1.13

The TSF shall support peer identifiers of the following types: [***IP address, Fully Qualified Domain Name (FQDN), Distinguished Name (DN)***] and [***no other reference identifier type***]. (TD0037 applied)

FCS_IPSEC_EXT.1.14

The TSF shall not establish an SA if the presented identifier does not match the configured reference identifier of the peer. (TD0037 applied)

FCS_IPSEC_EXT.1.15

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. (Renumbered per TD0037)

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

FCS_RBG_EXT.1.1

The [***TOE***] shall perform all deterministic random bit generation services in accordance with NIST Special Publication 800-90A using [***CTR_DRBG (AES)***]. (TD0079 applied)

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 [***256 bits***] of entropy at least equal to the greatest security strength (according to NIST SP 800-57) of the keys and hashes that it will generate.

5.1.2 User data protection (FDP)

5.1.2.1 Full Residual Information Protection (FDP_RIP.2)

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.

5.1.3 Identification and authentication (FIA)

5.1.3.1 Extended: Pre-Shared Key Composition (FIA_PSK_EXT.1)

FIA_PSK_EXT.1.1

The [***TOE***] shall be able to use pre-shared keys for IPsec.

FIA_PSK_EXT.1.2

The [***TOE***] shall be able to accept text-based pre-shared keys that:

- are 22 characters and [***[64 characters]***];
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: '!', '@', '#', '$', '%', '^', '&', '*', '(', and ')').

FIA_PSK_EXT.1.3

The [***TOE***] shall [***be able to [accept] bit-based pre-shared keys***].

5.1.3.2 Extended: X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1

The [***TOE***] 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 [***no other purpose***] 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***] 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.

5.1.3.3 Extended: X.509 Certificate Use and Management (FIA_X509_EXT.2)

FIA_X509_EXT.2.1

The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec exchanges, and [***no additional uses***].

FIA_X509_EXT.2.2

When a connection to determine the validity of a certificate cannot be established, the [***TOE***] shall [***accept the certificate***].

FIA_X509_EXT.2.3

The [***TOE***] shall not establish an SA if a certificate or certificate path is deemed invalid.

5.1.4 Security management (FMT)

5.1.4.1 Specification of Management Functions (FMT_SMF.1(1))

FMT_SMF.1(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**].

5.1.4.2 Specification of Management Functions (FMT_SMF.1(2))

FMT_SMF.1(2).1

The [***TOE, 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 this PP,
- ability to update the TOE, and to verify the updates,

- ability to configure all security management functions identified in other sections of this PP,
- ability to configure the reference identifier for the peer, (per TD0037)
- [***no other actions***].

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Extended: TSF Self Test (FPT_TST_EXT.1)

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***] 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 and hash for integrity].***

5.1.5.2 Extended: Trusted Update (FPT_TUD_EXT.1)

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.

5.1.6 Trusted path/channels (FTP)

5.1.6.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1

Refinement: The [***TOE***] 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.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM coverage

ATE: Tests	ATE_IND.1: Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability survey

Table 5-2 **Assurance Components**

5.2.1 Development (ADV)

5.2.1.1 Basic functional specification (ADV_FSP.1)

ADV_FSP.1.1d

The developer shall provide a functional specification.

ADV_FSP.1.2d

The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c

The functional specification shall provide rationale for the implicit categorisation of interfaces as SFR-non-interfering.

ADV_FSP.1.4c

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e

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

ADV_FSP.1.2e

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6c

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

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

5.2.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e

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

AGD_PRE.1.2e

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d

The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c

The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e

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

5.2.3.2 TOE CM coverage (ALC_CMS.1)

ALC_CMS.1.1d

The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c

The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e

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

5.2.4 Tests (ATE)

5.2.4.1 Independent testing - conformance (ATE_IND.1)

ATE_IND.1.1d

The developer shall provide the TOE for testing.

ATE_IND.1.1c

The TOE shall be suitable for testing.

ATE_IND.1.1e

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

ATE_IND.1.2e

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability survey (AVA_VAN.1)

AVA_VAN.1.1d

The developer shall provide the TOE for testing.

AVA_VAN.1.1c

The TOE shall be suitable for testing.

AVA_VAN.1.1e

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

AVA_VAN.1.2e

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Cryptographic support
- User data protection
- Identification and authentication
- Security management
- Protection of the TSF
- Trusted path/channels

6.1 Cryptographic support

The VPN client TOE provides data-in-transit (DIT) protections for all network communications by ensuring that all network traffic from the High-level operating system (Android) cell is encapsulated by the Outer DIT cell in an IPsec tunnel. The outer DIT cell is a Linux 3.10.84 kernel with strongSwan IPsec, OpenSSL cryptographic library, as well as other non-cryptographic supporting libraries.

All cryptographic operations are performed by the OpenSSL cryptographic library version 1.0.2f, using FIPS OpenSSL version 2.0.14 running in the Outer DIT cell. Refer to Table 6-1 for information about specific cryptographic algorithms used by the TOE and the CAVP certificates demonstrating compliance with corresponding standards.

Table 6-1 CAVP Algorithm Certificates

Algorithm	NIST Standard	SFR Reference	Provider	Cert#
AES: 128/256 CBC and GCM modes	FIPS 197, SP 800-38A/D	FCS_COP.1(1)	OpenSSL	AES: #4476

RSA: 2048
Sig(Gen) & SigVer
	FIPS 186-4
	Appendix B.3
		FCS_CKM.1(2)
		FCS_COP.1(2)
			OpenSSL	RSA: #2446

ECDSA:
KeyGen
SigGen & SigVer
P-256, P-384, P-521
	FIPS 186-4
	Appendix B.4
		FCS_CKM.1(2)
		FCS_COP.1(2)
			OpenSSL	ECDSA: #1093

SHA1, SHA-256, SHA-384, SHA-512	FIPS 180-4	FCS_COP.1(3)	OpenSSL	SHA: #3686
HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	FIPS 198-1 & 180-4	FCS_COP.1(4)	OpenSSL	HMAC: #2969
DSA: KeyPairGen(2048)	FIPS 186-4	FCS_CKM.1(1)	OpenSSL	DSA: #1200
CVL All SP800-56A except KDF: FFC & ECC	SP 800-56A, SP 800-56B	FCS_CKM.1(1)	OpenSSL	CVL: #1191
DRBG	SP 800-90A	FCS_RBG_EXT.1	OpenSSL	DRBG: #1456

The TOE is an IPsec VPN client which includes Critical Security Parameters (CSP) and/or keys to support this VPN functionality. The following table enumerates these CSP/keys, provides a brief statement describing their purpose, storage location as well as information about the clearing of these values.

References to crypto erase as a method of clearing refers to the deletion of the certificate by clearing the key used to encrypt the CSP. Once the key has been cleared, the data encrypted by it is considered to have been cryptographically erased. The reference to zero overwrite, indicates that the value being cleared is overwritten, byte-by-byte using a zero value and a single pass.

Table 6-2 CSP Identification and Clearing

CSP/Key Name:	Origin/Purpose:	Storage Location:	Cleared upon:	Type of clearing:
DH Group Parameters (supported DH groups)	RFC defined parameters hardcoded into the TSF/used in the ephemeral Diffie-Hellman key exchange	Executable Image in Flash	N/A – Public values	N/A – Public values
User IPsec X.509v3 Certs (RSA/ECDSA)	Entered by the user/used for client authentication	TOE Platform Keystore	On wipe function	Crypto erase
CA IPsec X.509v3 Certs (RSA/ECDSA)	Entered by the user/used to authenticate the gateway	TOE Platform Keystore	N/A – public values	N/A = public values
IKEv2 IKE_SA Encryption Keys (AES CBC or GCM)	Generated as part of IKEv2 IKE_SA establishment/used to encipher/decipher traffic	Memory/RAM	No longer needed by trusted channel	Zero overwrite
IKEv2 IKE_SA MAC Keys (HMAC-SHA)	Generated as part of IKEv2 IKE_SA establishment/used for traffic integrity	Memory/RAM	No longer needed by trusted channel	Zero overwrite
IKEv2 CHILD_SA Encryption Keys (AES CBC or GCM)	Generated as part of IKEv2 CHILD_SA establishment/ used to encipher/decipher traffic	Memory/RAM	No longer needed by trusted channel	Zero overwrite
IKEv2 CHILD_SA Keys (HMAC-SHA)	Generated as part of IKEv2 CHILD_SA establishment/ used for traffic integrity	Memory/RAM	No longer needed by trusted channel	Zero overwrite

The TOE implements IPsec and can operate in tunnel mode while being conformant with RFC 4301. The TOE supports AES-GCM-128 and AES-GCM-256, AES-CBC-128, and AES-CBC-256 modes for use with ESP. The AES-GCM ciphers are used in compliance with RFC 4106. The AES-GCM-128, AES-GCM-256, AES-CBC-128, and AES-CBC-256 ciphers can be used with either IKEv2 payloads. The TOE supports only IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23.

The TOE supports the above capabilities as specified by the following RFCs.

Table 6-3 IPsec RFCs

RFC	Title
2407	The Internet IP Security Domain of Interpretation for ISAKMP
2408	Internet Security Association and Key Management Protocol (ISAKMP)

2409	The Internet Key Exchange (IKE)
3602	The AES-CBC Cipher Algorithm and Its Use with IPsec
4106	The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)
4301	Security Architecture for the Internet Protocol
4303	IP Encapsulating Security Payload (ESP)
4307	Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)
4945	The Internet IP Security PKI Profile of IKEv1/ISAKMP, IKEv2, and PKIX
5282	Using Authenticated Encryption Algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) Protocol
5996	Internet Key Exchange Protocol Version 2 (IKEv2)
6379	Suite B Cryptographic Suites for IPsec

The TOE does not support direct editing of SPD rules. The TOE implements SPD rules that are defined implicitly through the configuration and connection of a VPN session. A VPN connection causes the TOE to implicitly define a PROTECT rule to IPsec encrypt and send all TOE traffic to the VPN gateway along with a DISCARD rule to reject any traffic not part of the established VPN connection. BYPASS is not supported as all traffic is directed through the VPN after provisioning.

The TOE is provisioned with Phase 1 and Phase 2 SA lifetime values which it proposes as initiator and other SA maximum values which it enforces as a responder. The Phase 1 SA proposal is 24 hours or less, and the phase 1 SA Max accepted value as a responder is 1 day (i.e., 24 hours). The Phase 2 SA proposal is 8 hours or less and the Phase 2 SA maximum lifetime accepted as a responder is 8 hours. As an initiator, the TOE sends the SA proposal value stated above. If the responder changes the value, the TOE will accept other values up to the SA maximum. As a responder, if the initiator’s SA lifetime is below the SA maximum, the TOE uses the proposed value, otherwise it offers the SA maximum (which the initiator must accept or reject).

The “x” value (256 bits) that is used in the IKE DH key exchange and the nonce (128 bits) are both obtained from the DRBG specified in FCS_RBG_EXT.1. This ensures that the probability they are repeated will be less than 2^256.

The TOE configures one DH group per profile. Connections can be made only using the configured DH group. The TOE supports Diffie-Hellman groups 14, 19, 20 and 24.

Table 6-4 Supported DH Groups

DH Group	Modulus	Strength
14	2048 bits MODP	112-bits
19	256-bit Random ECP	128-bits
20	384-bit Random ECP	192-bits
24	2048-bit MODP Group with 256-bit Prime Order Subgroup	112-bits

The TOE performs peer authentication using an RSA x509v3 certificate, or an ECDSA x509v3 certificate. The x509v3 certificates must be conformant with RFC 4945. When certificates are used for authentication, the TOE establishes an SA only if the IP address, FQDN or Distinguished Name (DN) contained in a certificate matches the expected IP address, FQDN or Distinguished Name (DN) configured for the profile. Additional checks on a certificate enforce proper key usage, the validity period and revocation status. The TOE does not generate certificate requests, but rather requires certificates to be loaded through the platform. The TOE can also authenticate IPsec peers using pre-shared keys, using keys as specified by FIA_PSK_EXT.1 described in Section 6.3..

The TOE ensures that IKEv2 IKE_SA connections use the same algorithm and key size as is used by IKEv2 CHILD_SA connections. It is not possible to configure different algorithms for an IKEv2 IKE_SA and CHILD_SA connections.

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1(1): The OpenSSL library in the TOE generates ECDSA asymmetric keys used for key establishment. Refer to Table 6-1 for the corresponding CAVP certificate demonstrating compliance with required algorithms and standards.
- FCS_CKM.1(2): The OpenSSL crypto library generates asymmetric cryptographic keys for use with IKE peer authentication. Refer to Table 6-1 for the corresponding CAVP certificate demonstrating compliance with required algorithms and standards.
- FCS_CKM_EXT.2: The TOE platform provides key storage. The Android Key Store is used by the Android Cell, and is accessible to Android applications. The Block Cell also stores keys on behalf of the Outer DIT VPN client and is accessible through the HSM-Proxy Cell only. The Outer DIT cell calls the HSM proxy cell to use the Block cell to store its keys.
- FCS_CKM_EXT.4: Refer to Table 6-2 above for a list of the CSPs, their storage location and clearing approaches.
- FCS_COP.1(1): The OpenSSL library provides an implementation of AES that operates in both CBC and GCM modes, using key lengths of 128-bit and 256-bit. Refer to Table 6-1 for the corresponding CAVP certificate demonstrating compliance with these algorithms.
- FCS_COP.1(2): The OpenSSL library provides cryptographic signature services for RSA and ECDSA signature generation and verification. ECDSA operations use NIST curves P-256, P-384 and P-521. Refer to Table 6-1 for the corresponding CAVP certificate demonstrating compliance with these algorithms.
- FCS_COP.1(3): The OpenSSL library provides an implementation of the SHA-1, SHA-256, SHA-384 and SHA-512 hashing algorithms. Refer to Table 6-1 for the corresponding CAVP certificate demonstrating compliance with these algorithms. These hash functions can be defined for use in an IPsec connection.
- FCS_COP.1(4): The OpenSSL library provides an implementation of a keyed-hash message authentication code (HMAC). The library provides the HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512 algorithms. Refer to Table 6-1 for the corresponding CAVP certificate demonstrating compliance with these algorithms. These keyed-hash functions can be defined for use in an IPsec connection. The HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512 algorithms are used with key sizes and block sizes of 160, 256, 384 and 512 respectively, producing output MAC lengths equal to the block size.
- FCS_IPSEC_EXT.1: The TOE implements IPsec in accordance with FCS_IPSEC_EXT.1 as described throughout section 6.1 above.
- FCS_RBG_EXT.1: The TOE utilizes the DRBG provided by the OpenSSL cryptographic library for the generation of all random values used as keys, nonces and salts. The OpenSSL library provides an SP 800-90A compliant AES-256 CTR-DRBG that is seeded using 384-bits from the hw_drbg provided by the MSM8952 SoC. Refer to Table 6-1 for the corresponding CAVP certificate.

6.2 User data protection

The TOE ensures that no residual information exists in network packets. The TOE encapsulation/de-encapsulation mechanisms ensure that any encapsulated data only contains cipher text and any required padding is set to a deterministic value. When the TOE allocates a new buffer for either an incoming or outgoing network packet, the new packet data will be used to overwrite any previous data in the buffer. If an allocated buffer exceeds the size of the packet, additional space is overwritten (padded) with zeros before the packet is forwarded (to the external network or delivered to the appropriate, internal application).

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_RIP.2: The TOE ensures that previous information contents of resources used for new objects are not discernible in any new object, such as files, network packets, as described above.

6.3 Identification and authentication

The TOE uses X.509 certificates for authentication. The TOE requires that for each VPN connection, the user specify the client certificate the TOE will use (the user must have previously loaded such a certificate into the keystore) and specify the CA certificate to which the server's certificate must chain. The TOE then uses the specified certificate when attempting to establish that VPN connection. The TOE validates authentication certificates (including the full path) and checks their revocation status using either a CRL or OCSP request, based upon the contents of the certificate. The TOE processes a VPN connection to a server by first comparing the Identification (ID) Payload received from the server against the certificate sent by the server, and if the IP address or FQDN of the certificate does not match the ID, then the TOE does not establish the connection. Assuming the server's certificate matches the ID, the TOE then validates that it can construct a certificate path from the server's certificate through any intermediary CAs to the CA certificate specified by the user in the VPN configuration. If the TOE can successfully build the certificate path, then the TOE will next check the validity of the certificates (e.g., checking its validity dates and that the CA flag is present in the basicConstraints extension for all CA certs). Assuming the certificates are valid, the TOE finally checks the revocation status of all certificates (starting with the server's certificate and working up the chain). The TOE will accept any certificate for which it cannot determine the validity and establish the connection. Section 6.1 describes additional details of how the TOE uses certificates in its IPsec architecture.

The TOE can also authenticate IPsec peers using pre-shared keys. A pre-shared key can be between 1 and 64 characters, composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")"). The TOE can also accept bit-based pre-shared keys, when specified in hex format

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_PSK_EXT.1: The TOE supports the use of pre-shared IPsec keys used to create IPsec connections. The pre-shared keys can be composed as required and as described above.
- FIA_X509_EXT.1: This requirement is satisfied by the TOE, which performs all needed certificate validation (including the validation of certificate fields, its path, and its revocation status). Refer to the above material for more information.
- FIA_X509_EXT.2: The TOE uses X.509v3 certificates for authentication in IPsec exchanges. The D4 Secure VPN Client will always accept certificates that cannot be determined to be valid (i.e., cannot determine revocation status).

6.4 Security management

The following security management functions are provided directly by the TOE and/or implemented in the TOE Platform as indicated below:

- The TOE provides functions allowing the user to select VPN gateway and credentials used to connect to those gateways (i.e., ability to configure the reference identifier for the peer).
- The TOE platform provides the ability to load X.509v3 certificates used for VPN connections using IPsec.
- The TOE provides the ability to configure the version of IKE protocol that is to be used for communication with a given VPN gateway.
- The TOE provides the ability to configure the IKE authentication techniques to be used for communication with a given VPN gateway.
- The TOE provides the ability to configure the crypto-period for the established session keys. The unit of measure for configuring the crypto-period is one hour.
- The TOE always accepts a certificate when the certificate revocation check cannot be performed.
- The TOE provides the ability to specify the algorithm suites that may be proposed and accepted during the IPsec exchanges.
- The TOE provides the ability to configure all security management functions identified in other sections of this ST.

- The TOE Platform provides the ability to update the TOE, and to verify the updates.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_SMF.1(1): The TOE provides the functions necessary to specify VPN gateways and the corresponding credentials used to establish VPN connections as described above.
- FMT_SMF.1(2): The TOE, and TOE platform provide the functions necessary to manage the security functions described in this security target as described above.

6.5 Protection of the TSF

The TOE performs known answer power on self-tests (POST) on its cryptographic algorithms to ensure that they are functioning correctly. The TOE executes the OpenSSL Library known-answer tests on the OpenSSL cryptographic functions to ensure they are working correctly. These tests cover the following Cryptographic Algorithm Tests:

- AES-CBC, AES-GCM Known Answer Test
- HMAC-SHA-1 Known Answer Test
- HMAC-SHA-256 Known Answer Test
- HMAC-SHA-384 Known Answer Test
- HMAC-SHA-512 Known Answer Test
- SHA-1 Known Answer Test
- SHA-256 Known Answer Test
- SHA-384 Known Answer Test
- SHA-512 Known Answer Test
- AES-CTR DRBG Known Answer Test
- RSA Known Answer Test
- ECDSA Known Answer Test

In the event of a self-test failure, the OpenSSL library will enter an error state and a specific error code will be returned indicating which self-test or conditional test has failed. The OpenSSL library will not provide any cryptographic services while in this state.

The TOE invokes these self-tests of the OpenSSL library when the library is initialized to ensure that those cryptographic algorithms are working correctly.

The TOE platform supports an update process that will “side-loading” an update via USB tethering. Since the Outer DIT VPN client is built-in to the TOE platform, it is updated only with a TOE platform firmware update. A firmware update occurs if a new image is detected when the provisioning service detects a new image during the device boot process. When a new image is detected and before the new image is installed, the HTC bootloader validates the new Cog[2] Secure Boot image in the same manner in which it validates an already installed boot image. This ensures that the new image has been signed using the same RSA 2048-bit HTC public key by which the bootloader was signed (i.e., a public key whose hash is burned into the device’s fuses).Once installed, an update entirely replaces the Cog Secure Boot image, and the Hypervisor system image (which includes cell images), then restarts the boot process. The normal boot process then re-validates every part of the boot chain.

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_TST_EXT.1: The TOE executes the self-test identified above upon the first invocation of the OpenSSL library (typically upon power-on).
- FPT_TUD_EXT.1: The TOE Platform's user interface provides a method to query the current version of the TOE Platform software/firmware (including the hypervisor version, and kernel version) and hardware (model and version). Because the VPN client is built-in to the TOE platform, this same version

[2] Cog Systems is one of the developers. They have used their own name to identify part of the boot-loading process software.

information specifies the VPN TOE version and the TOE Platform update mechanism is used to install VPN TOE updates (because updating the VPN TOE is an update to the TOE Platform). The TOE Platform utilizes a digital signature mechanism to verify TOE Platform updates. Failed verification causes the update to be rejected and a security warning to be displayed on the device.

6.6 Trusted path/channels

Refer to section 6.1 for a description of how the TOE can establish IPsec VPN connections with configured VPN gateways. The resulting VPN(s) ensure that both ends of the channel are authenticated and the channel protects data from disclosure and modification.

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1: The TOE initiates all communication with a VPN Gateway using an IPsec VPN. The TOE key exchange uses IKEv2. The TOE uses IPsec to provide assurance in the identification of endpoints and to protect transmitted data from disclosure and modification.